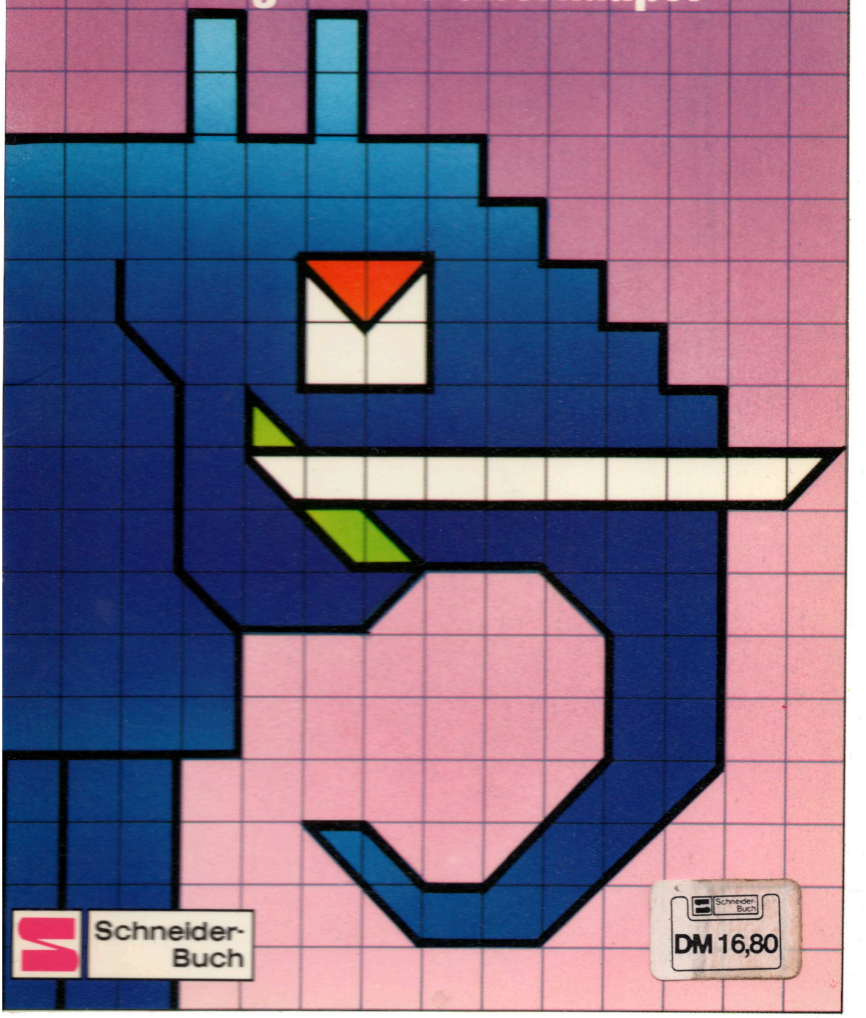
Computer kinderleicht

Wie geht das überhaupt?



Schneider-Buch

Schneider Buch

DM 16,80

Computer kinderleicht

Wie geht das überhaupt?

Georg Seeßlen
Peter Fritzsche

Schneider-Buch

Inhalt

Vorwort 5

Die Geschichte vom Computer-Max 9

Wir erobern den Bildschirm 20

Was heißt hier PRINT? 56

Quizmaster, Codeknacker und Gedächtniskünstler 93

Tips und Tricks für das Eingeben der Listings 111

Listings der Spiele von Band 1 113

- Liste der Steuerzeichen 114
- Liste der BASIC-Befehle 116
- Gebrauchsanweisung für Memory 137
- Platz für eigene Notizen 155

Vorwort

Irgendwann kommt für jeden der Einstieg in die neue und auf den ersten Blick so verwirrend vielfältige Welt der Computerei. Ob in der Schule oder zu Hause am Heimcomputer – der Umgang mit Computern wird für die nächsten Generationen von Kindern so selbstverständlich sein wie heute noch Bücher lesen, Musik hören oder Filme sehen.

Selbst wenn man die ersten Schritte an der Hand von Lehrern, Freunden, Eltern oder Geschwistern macht – was wirklich im Computer steckt, ist nur durch Bücher zu erfahren. Bücher, die den Leser zu eigenem Experimentieren, eigenem Ausprobieren, eigener Computer-Phantasie führen.

Die Reihe „Computer kinderleicht" des Franz Schneider Verlags bietet eine systematische und unterhaltsame Einführung in sämtliche Bereiche des Umgangs mit Computern für Mädchen und Jungen.

Dabei versprechen wir keineswegs, gleich „Alles über . . ." zu erfahren. Vielmehr wird in klar gegliederten und überschaubaren Lernabschnitten der Computer als kreatives Medium erlebt. Durch Spiel und Übung eignen sich die Kinder das nötige Wissen selbst an und lernen dabei auch den Computer als „Partner" bei Lernspielen und als willkommene Unterstützung kennen, der bei der Bewältigung des schulischen Lernstoffes hilft.

Die Bände der Reihe „Computer kinderleicht", geschrieben von erfahrenen Pädagogen, sind wirkliche Kinderbücher, weil Computer-neugierige und

schließlich Computer-begeisterte Kinder mit ihrer Kritik und ihren Vorschlägen an der Gestaltung beteiligt waren.

Das heißt auch: Die Bücher lassen sich von den Kindern eigenständig und ohne Hilfe von Erwachsenen benutzen. Jede Frage wird dann beantwortet, wenn sie beim eigenen Probieren am Gerät auftaucht. Kompliziertes Computer-Chinesisch wird vermieden. Allerdings werden alle gebräuchlichen Fachausdrücke dann erklärt, wenn sie im Zusammenhang mit eigenen Erfahrungen wichtig sind.

Auf Phasen, in denen Neues dazugelernt wird, folgen immer solche, in denen die Kinder das Gelernte spielerisch anwenden können. Aus einem simplen „Rechner" wird dabei ein Gerät, das immer wieder neue Anwendungen ermöglicht. Auch solche, von denen sich die Handbücher nichts träumen lassen.

Die Kinder lernen den Computer „ganzheitlich" kennen; Rechner und Bildschirm werden zu einem Experimentierfeld, das auch solche Kinder zum Spielen und Lernen anregt, die neben dem Computer noch viele andere Interessen haben. „Computer kinderleicht" – das heißt Computerei erlernen ohne Mißerfolge, ohne „schwarze Löcher" und ohne Langeweile.

Alle Bücher bieten als zusätzlichen Anreiz etliche Lernspiele, die auf die eigenständigen Computer-Erfahrungen zugeschnitten sind. Lustige und spannende Lern- und Übungshilfen zum Rechnen, Schreiben und Kombinieren, bei denen auch die „Action" nicht zu kurz kommt.

Diese Spiele sind als Listings im Buch enthalten (zum Abtippen für geübtere Eltern oder Geschwister); sie sind wahlweise aber auch als Datenträger

(auf Diskette) zusätzlich zum Buch erhältlich. Um sie problemlos in den Computer zu laden, sind keine Vorkenntnisse nötig.

Die Bücher sind zum Ausprobieren auf einem Commodore 64 geschrieben. Die ersten beiden Bände lassen sich auch mit dem VC 20 verwenden. Die Spielprogramme sind allerdings nur auf dem Commodore 64 lauffähig.

Die einzelnen Bände der Reihe „Computer kinderleicht" bauen jeweils aufeinander auf. Man kann sie jedoch – mit Hilfe des Handbuchs und aufgrund eingestreuter „Wiederholungen" – auch einzeln verwenden.

Jeder Band hat einen thematischen Schwerpunkt, ohne daß dabei die „Ganzheitlichkeit" und die beim experimentierenden Lernen unabdingbaren Seitenlinien vernachlässigt würden.

Band 1 macht mit dem Computer und der Sprache BASIC vertraut. Ziel ist, die Tastatur und die bildhafte Umsetzung durch die Spielvorschläge zu beherrschen. Das sollte die Voraussetzung sein, damit beim eigentlichen Programmieren dann nicht allzu viel Zeit und Lust durch wiederkehrende „kleine" Fehler verlorengeht.

Zugleich werden erste Operationen mit BASIC ausprobiert und bereits in eigene Spiele umgesetzt.

Band 2 ist vor allem grundlegenden Zahlenoperationen gewidmet. Er zeigt den Kindern, wie man mit einfachen Mitteln den Computer als Unterstützung bei Rechenübungen verwenden kann, ohne daß der Computer zum „Mogeln" verwendet wird.

Zugleich werden erste logische Verknüpfungen

und die Anwendung von Variablen ausprobiert.

Nach diesem Band mit dem Schwerpunkt Rechnen folgt

Band 3 (in Vorbereitung) mit dem Schwerpunkt Text. Es werden Grundlagen der Textverarbeitung in einfachen und wiederum spielhaften Beispielen vorgestellt. Eines der beiliegenden Programme bietet die Möglichkeit, Sprachtraining mit dem Computer zu betreiben.

Zugleich werden im Bereich von BASIC Erfahrungen mit neuen Operationen ermöglicht. Ein Schwerpunkt liegt hier bei den String-Variablen und bei Text-Abfragen.

Band 4 (in Vorbereitung) ist das erste auf Anwendungen konzentrierte Buch der Reihe. Es ist nur in Verbindung mit dem Datenträger erhältlich und bietet ein Spiel als kombiniertes Grafik- und Text-„Abenteuer".

In diesem Buch werden zwar keine neuen BASIC-Operationen erläutert, doch führen auch Auswahl und Menüsteuerung weiter in die Welt der Computerei ein.

Weitere Bände in Vorbereitung

Die Geschichte vom Computer-Max

Vor einiger Zeit saßen wir, Beate, Peter und Georg, im Wohnzimmer und spielten an unserem Computer herum.

Die schönsten Spiele, die tollsten Rechenprogramme, Musik und Bildschirmtricks sind auf diese Weise entstanden. Einfach durch viel Ausprobieren, durch Spielen mit dem Computer.

Dabei ist es ganz egal, ob man schon ein ausgefuchster Hacker ist oder gerade erst mit dem Computer-Abenteuer angefangen hat. Ausprobieren ist das ganze Geheimnis.

Je mehr wir nämlich ausprobieren, desto besser lernen wir unseren Computer kennen. Und der nimmt es überhaupt nicht übel, wenn wir dabei auch Fehler machen und gar nichts klappen will. Computer sind unendlich geduldig und schimpfen nie.

Bald wirst du mit deinem Computer schon etwas „sprechen" können. Mit einem normalen Heimcomputer kann man natürlich nicht einfach so sprechen, wie man es vielleicht mit den Robotern könnte, die wir aus dem Kino kennen.

Wir sprechen erst einmal schriftlich miteinander. Wir, die Leute, die mit dem Computer spielen und arbeiten wollen, tippen auf der Tastatur Nachrichten und Aufgaben für den Computer. Die werden dann auf dem Bildschirm sichtbar. Und der Computer antwortet ebenfalls auf dem Bildschirm. Entweder löst er die Aufgaben, die wir ihm gestellt haben. Oder er sagt uns, so gut er eben vermag, warum er uns keine sinnvolle Antwort geben kann.

Damit wir uns mit dem Computer verständigen können, brauchen wir die Tastatur. Sie sieht fast so aus wie bei einer Schreibmaschine. Auf dieser Tastatur tippen wir ein, was wir vom Computer wollen.

Wir schreiben dem Computer sozusagen einen Brief. Darin steht, was er für uns tun soll: eine Rechnung lösen zum Beispiel oder etwas zeichnen. Sogar ein Fragespiel oder auch eine bewegte Szene können wir den Computer auf den Bildschirm bringen lassen und noch vieles mehr.

Unseren Brief liest der Computer in Windeseile. Fast ebenso schnell führt er die Befehle aus, die wir ihm eingegeben haben. Das heißt, wenn er uns verstanden hat. Denn da ist er ziemlich stur. Der Computer ist schließlich nur eine Maschine und kein Mensch. Menschen können sich vielleicht überlegen, was gemeint ist, auch wenn es nicht ganz richtig ausgedrückt war. Der Computer kann das nicht – noch nicht.

Die Computer haben ihre eigenen sogenannten „Maschinensprachen", wie sich das für Maschinen gehört. Für uns sind diese Sprachen sehr schwer zu lernen. Damit wir leichter mit diesen Maschinen

reden können, hat man andere Sprachen entwickelt, solche, die von den Maschinen *und* von Menschen verstanden werden.

Diese Sprachen lassen sich ebenso einfach lernen wie etwa Englisch oder Italienisch. Sie heißen PASCAL, BASIC, LOGO, FORTRAN und so weiter.

In diesem Buch beschäftigen wir uns mit der Computersprache BASIC. Diese Sprache verstehen alle Heimcomputer, und deiner natürlich auch.

Wie bei allen Sprachen kommt es auch bei der Computersprache BASIC nicht allein aufs Auswendiglernen an, sondern vielmehr auf die Übung.

So ganz darf man allerdings die Computersprache BASIC mit einer „richtigen" Sprache nicht vergleichen. Englisch, Deutsch oder Französich haben ja eine riesige Zahl von Wörtern, Regeln und Ausnahmen von Regeln. Wenn wir dagegen von unse-

rem BASIC auch nur zehn bis zwanzig Wörter oder „Ausdrücke“ kennen, können wir mit dem Computer schon allerhand anstellen.

Dein Computer ist mit einem Bildschirm verbunden. Darauf siehst du, was du dem Computer geschrieben hast und was er dir antwortet – jedenfalls, wenn er verstanden hat, was du willst.

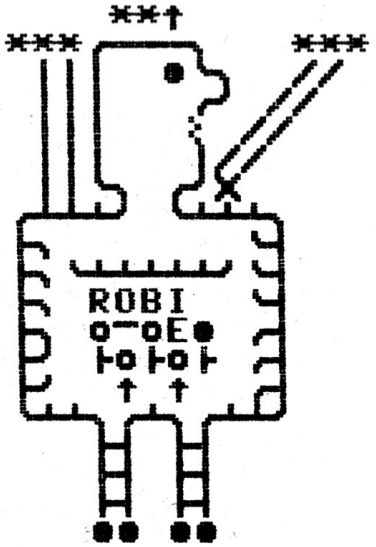
Das erinnert mich daran, daß ich von dem Nachmittag erzählen wollte, an dem Beate, Peter und ich im Wohnzimmer saßen und an unserem Computer herumprobierten.

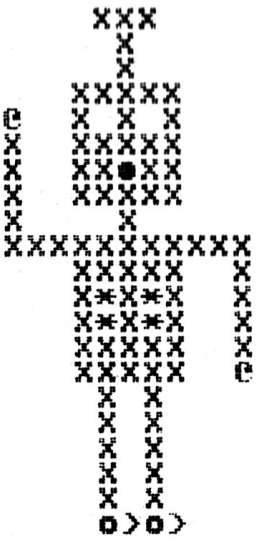
Wir wollten eine Figur mit dem Computer herstellen, also auf dem Bildschirm „zeichnen“. Diese Figur sollte uns auf unserer Reise in die Computerwelt begleiten, ein bißchen lustig sein und sich später auch bewegen lassen. Und alle Kinder, die unsere Computer-Bücher lesen würden, sollten das mit unserer Figur bald selber auf ihrem eigenen Computer können.

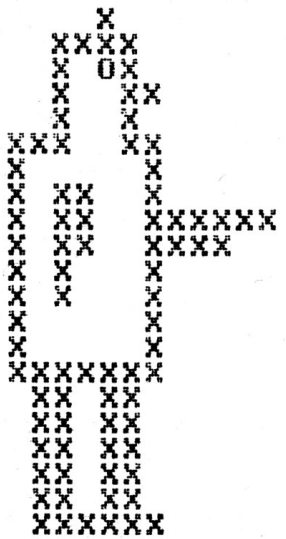
Vor allem aber sollte uns diese Figur auch etwas

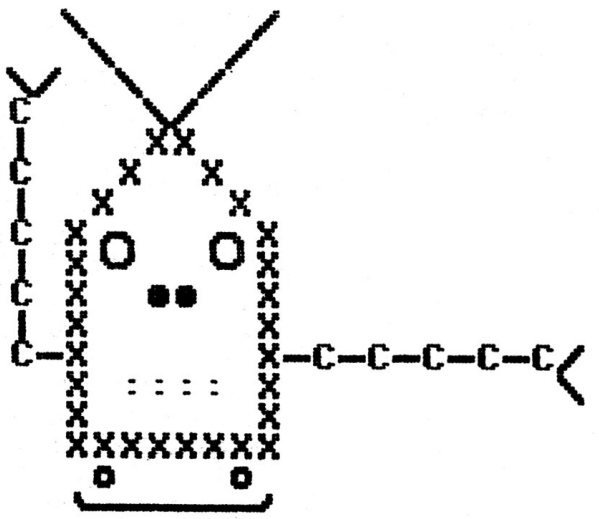
darüber erzählen, wie ein Computer „denkt“.

Zuerst entwarfen wir jede Menge Roboter: große und kleine, dünne und dicke. Mit Robotern, dachten wir, läßt es sich am besten zeigen, was in einem Computer so vor sich geht. Denn Roboter sind ja eigentlich nichts anderes als Computer mit einem Blechkörper drumherum, mit ein paar Armen und Greifwerkzeugen und mit Rollen oder Beinen unten dran, auf denen sie sich bewegen. Auf dem Bildschirm sahen sie so aus:









Aber mit der Zeit merkten wir, daß uns ein anderer Begleiter lieber wäre. Freilich gibt es sehr nette Roboter. Wer kennt nicht die beiden aus dem „Krieg der Sterne“:

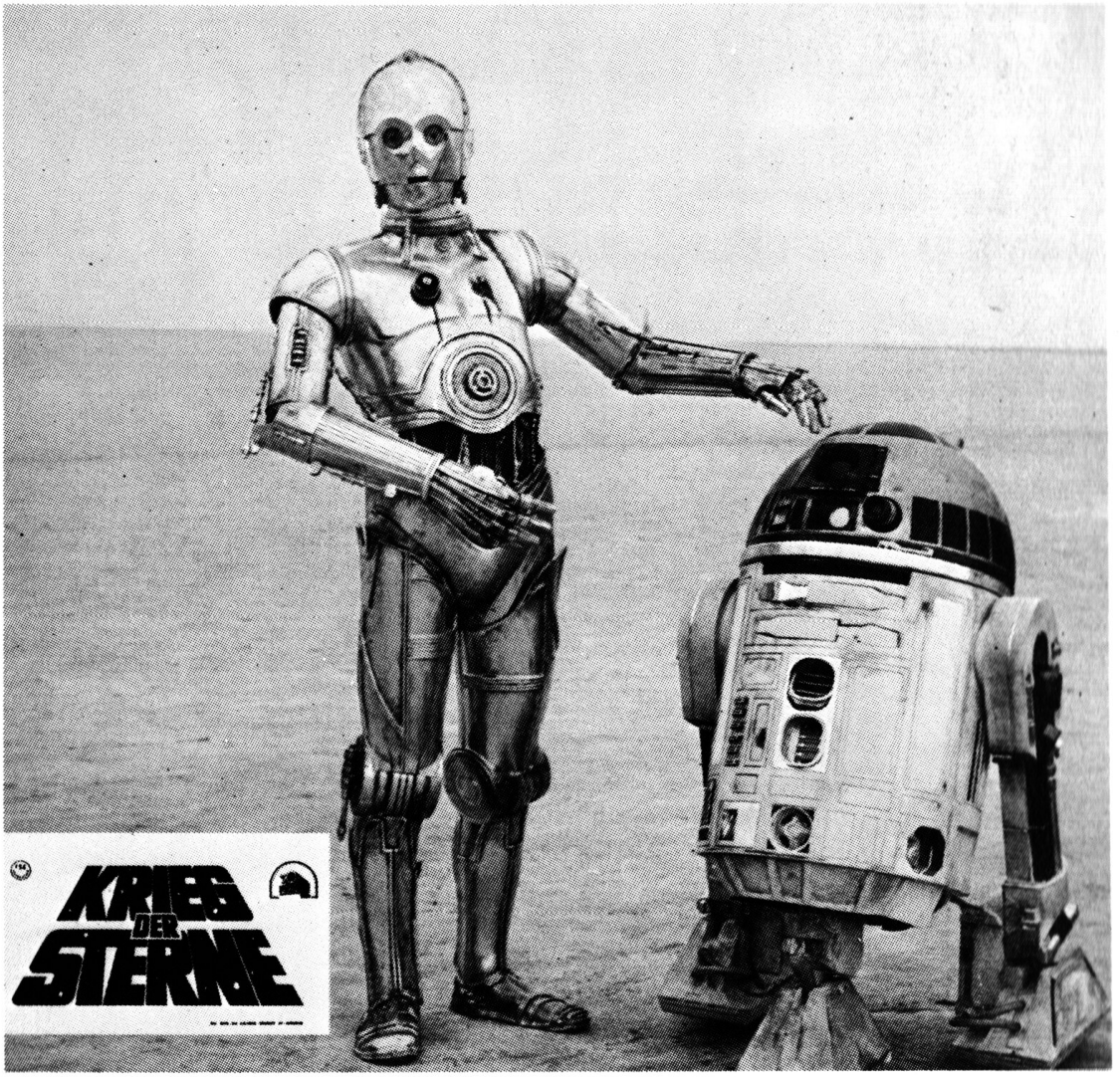


Foto: Lucasfilm Ltd. (LFL) 1983

Trotzdem, wir wollten ein richtig „lebendiges“ Wesen, ein Tier vielleicht. Wie wäre es zum Beispiel mit einer Schildkröte; so könnte sie aussehen:

Die Geschichte vom Computer-Max

Vor einiger Zeit saßen wir, Beate, Peter und Georg, im Wohnzimmer und spielten an unserem Computer herum.

Die schönsten Spiele, die tollsten Rechenprogramme, Musik und Bildschirmtricks sind auf diese Weise entstanden. Einfach durch viel Ausprobieren, durch Spielen mit dem Computer.

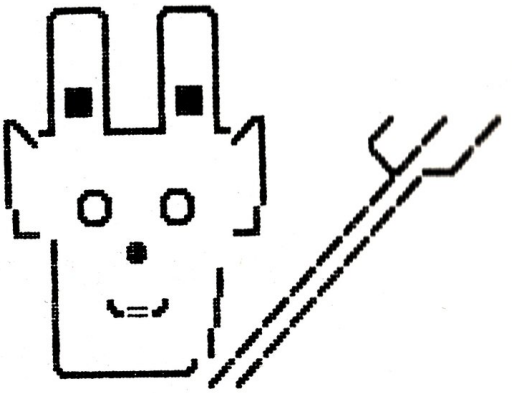
Dabei ist es ganz egal, ob man schon ein ausgefuchster Hacker ist oder gerade erst mit dem Computer-Abenteuer angefangen hat. Ausprobieren ist das ganze Geheimnis.

Je mehr wir nämlich ausprobieren, desto besser lernen wir unseren Computer kennen. Und der nimmt es überhaupt nicht übel, wenn wir dabei auch Fehler machen und gar nichts klappen will. Computer sind unendlich geduldig und schimpfen nie.

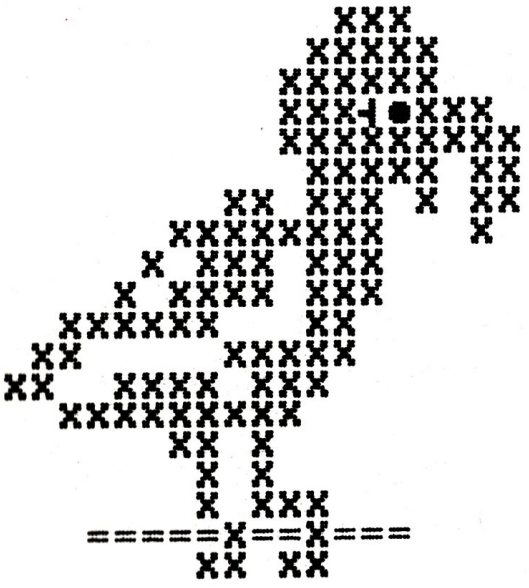
Bald wirst du mit deinem Computer schon etwas „sprechen" können. Mit einem normalen Heimcomputer kann man natürlich nicht einfach so sprechen, wie man es vielleicht mit den Robotern könnte, die wir aus dem Kino kennen.

Wir sprechen erst einmal schriftlich miteinander. Wir, die Leute, die mit dem Computer spielen und arbeiten wollen, tippen auf der Tastatur Nachrichten und Aufgaben für den Computer. Die werden dann auf dem Bildschirm sichtbar. Und der Computer antwortet ebenfalls auf dem Bildschirm. Entweder löst er die Aufgaben, die wir ihm gestellt haben. Oder er sagt uns, so gut er eben vermag, warum er uns keine sinnvolle Antwort geben kann.

So sieht unser Programmteufel tatsächlich aus



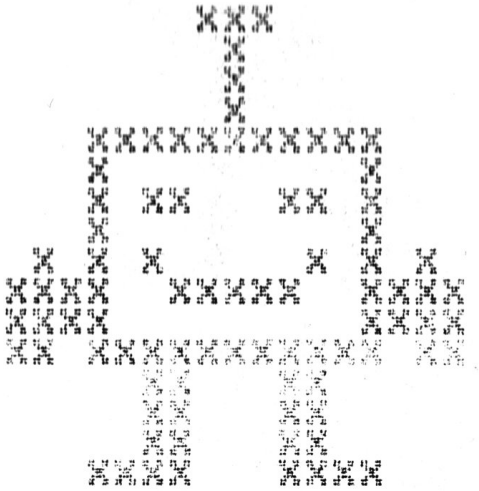
Und so auf dem Computerbildschirm



Unser Wunschbegleiter in die Computerwelt war noch nicht dabei, aber ein paar von diesen Wesen werden später wieder auftreten.

Eines nannten wir den „Programmteufel". Er wird dann auftauchen, wenn etwas schiefgehen kann. (Für ein späteres Buch dieser Reihe haben wir auch ein Computerspiel mit ihm entwickelt.)

Dann zeigte uns Peter seinen „Teli". „Teli" war ein lustiges TV-Männlein und sah so aus:



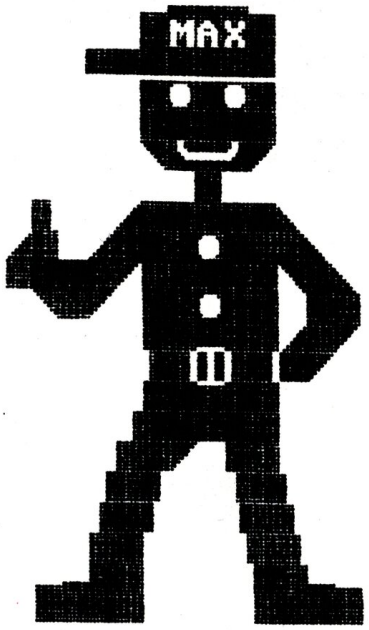
Allerdings war er etwas unbeweglich. Außerdem ließ er sich nicht mit jedem Computer gleich gut darstellen, so nahmen wir von „Teli" wieder Abschied.

Mittlerweile war es draußen dunkel geworden, und unseren Computer-Begleiter hatten wir immer noch nicht gefunden. Draußen regnete es in Strömen. Wir zündeten ein Feuer im Ofen an. Die Kinder kamen herein und setzten sich zu uns.

Ein bißchen probierten wir noch herum. Eigentlich wollten schon alle den Computer ausschalten, einen Tee trinken und eine Runde Mensch-ärgere-dich-nicht spielen.

Da stand unser Begleiter plötzlich auf dem Bildschirm! Das Kerlchen, das ziemlich frech unter sei-

ner Mütze hervorsah, war offenbar quicklebendig und dennoch ein Computer-Geschöpf.



Sobald wir ihm außerhalb des Bildschirms begegnen, sieht er so aus:

Schnell hatte er auch einen Namen: Computer-Max. Wir haben an diesem Abend doch noch Mensch-ärgere-dich-nicht gespielt. Dabei hatten wir das Gefühl, uns würde jemand zuschauen.

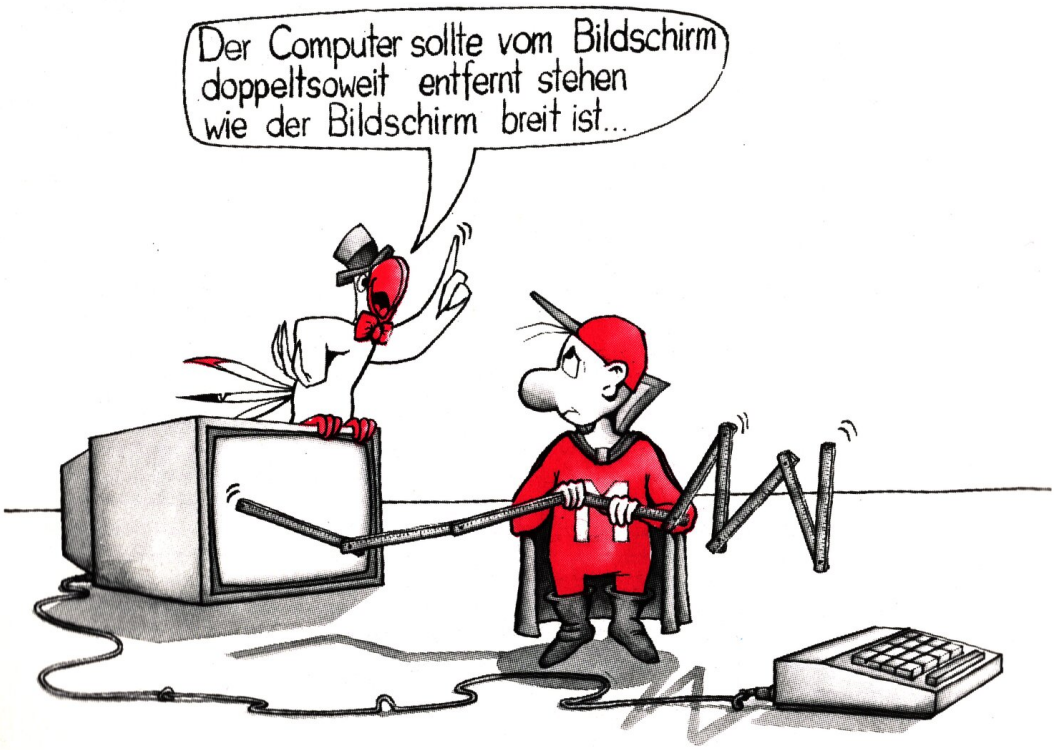
Vor lauter Begeisterung hatten wir nämlich vergessen, den Computer-Bildschirm auszuschalten.

Wir erobern den Bildschirm

Wahrscheinlich hast du mit Hilfe des mitgelieferten Handbuchs schon ein wenig an deinem Computer herumprobiert. Vielleicht haben dir deine Eltern, Geschwister oder Freunde dabei geholfen.

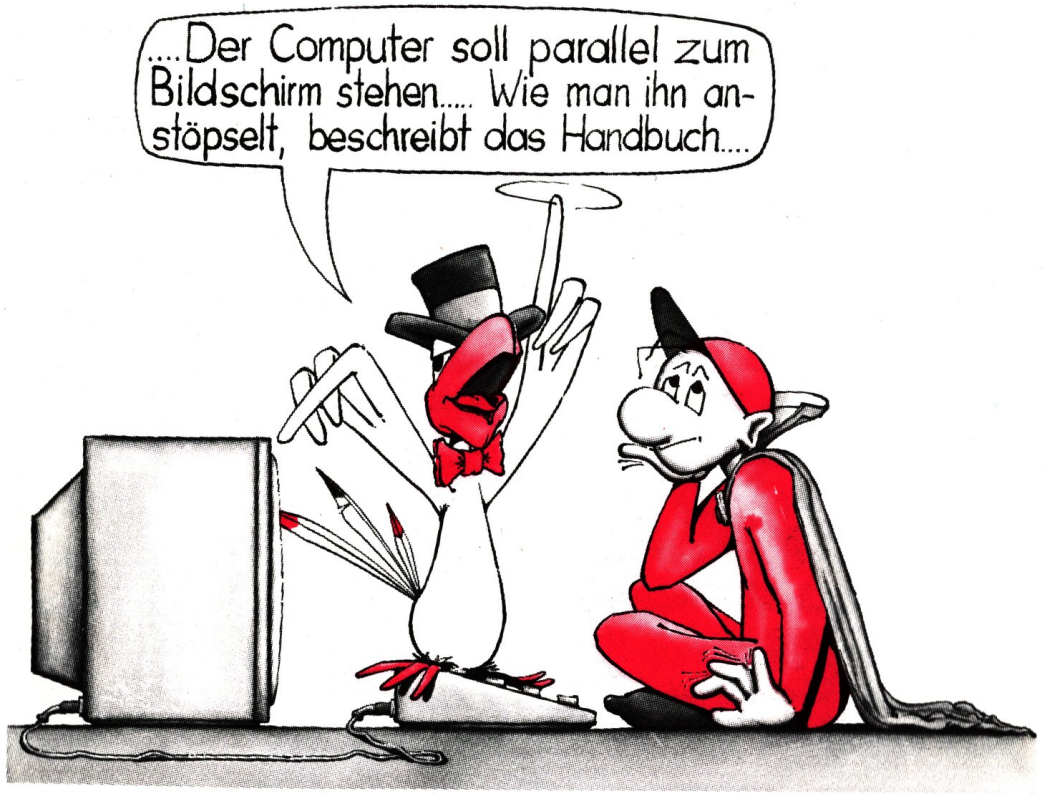
Unser Buch verstehst du auch dann, falls du jetzt das erste Mal vor einem Computer sitzt.

Natürlich kannst du vor deinem Computer auch stehen, hocken oder gar liegen. Der Computer soll nicht allzu weit vom Bildschirm entfernt sein – ungefähr doppelt so weit, wie der Bildschirm breit ist. Bildschirm und Computer müssen so stehen, daß man beide gut überblicken kann – damit man nicht um die Ecke schauen (und denken) muß.



Wie der Computer angestöpselt und eingestellt wird, entnimmst du am besten deinem Handbuch. Das ist nämlich von Computer zu Computer verschieden. *Noch besser ist es wohl, du läßt dir dabei zu Beginn von den Eltern oder Geschwistern helfen.*

Achte aber in jedem Fall darauf, daß die Kabel vom Computer zum Fernseher, zum Kassettenrecorder beziehungsweise zur Diskettenstation immer nur dann ein- und ausgesteckt werden sollen, wenn der Computer ausgeschaltet ist.

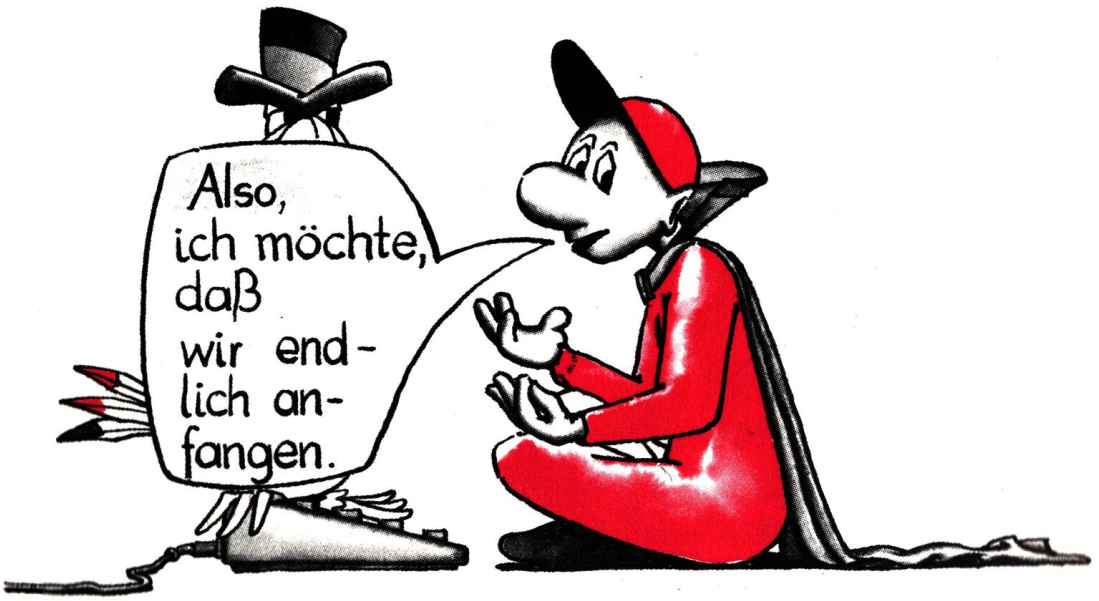


Für Computer gibt es eigene Bildschirme; sogenannte Monitore. Aber mit einem normalen Fernseher, am besten einem Farbfernseher, geht es auch. Dazu mußt du den Computer einschalten und am ebenfalls eingeschalteten Fernseher noch den

richtigen Kanal einstellen. Dann kann der Computer uns seine „Antwortbriefe“ schreiben. Den „Computerkanal“ stellt man ebenso ein wie ein bestimmtes Fernsehprogramm. Spätestens hier aber solltest du deine Eltern zu Rate ziehen, schon damit sie sich nicht am nächsten Sonntag ärgern, wenn sie ihren Lieblingskrimi oder die Fußballübertragung nicht reinkriegen.

Sofern der Fernseher genügend Programmknöpfe hat und der Kanal einmal gefunden ist, bleibt unser Kanal ein für allemal für den Computer reserviert. Nun darfst du sicher den Computer immer selbst anschließen.

Falls auf eurem Dachboden noch ein alter Schwarzweißfernseher steht – der ist wie geschaffen für das Computern. Auf Farbeffekte mußt du dann zwar verzichten. Aber auch ohne die Farbeffekte gibt es viele Möglichkeiten, mit dem Computer zu experimentieren – und der Rest der Familie kann unbehelligt die Weltmeisterschaft im Eistanzen verfolgen.



EINER VON UNS sitzt schon ganz ungeduldig mit seinem Computer vor dem Bildschirm. EINER VON UNS will jetzt keine langen Erklärungen mehr hören. EINER VON UNS will jetzt endlich loscomputern.

Hab noch ein wenig Geduld, lieber Max.

Sobald unser Computer richtig eingestellt ist, erscheint auf dem Bildschirm des Fernsehers ein rechteckiges Feld. Wie groß das ist, und wieviel Zahlen und Buchstaben da hineinpassen, hängt vom jeweiligen Computer-Typ ab.

Ganz oben im Bildschirm steht der „Name" deines Computers. In der Zeile darunter folgt eine ziemlich hohe Zahl, dahinter die beiden englischen Wörter BYTES FREE (sprich *"beits frie"*). Beim Commodore 64 zum Beispiel sieht das so aus:

38911 BYTES FREE

Diese Zahl (38911) sagt etwas darüber aus, wieviel Platz dein Computer hat, um sich all das zu merken, was er für dich tun soll. Da steht also, wie groß das **GEDÄCHTNIS** deines Computers ist. Aber egal, was für eine Zahl da steht, das Gedächtnis deines Computers ist groß genug für alles, was wir in diesem Buch miteinander probieren und ziemlich sicher auch für das, was du selber für ihn erfindest.

Wichtiger ist im Augenblick, was danach auf dem Bildschirm erscheint, nämlich rechts darunter das Wort

READY.

und darunter ein kleines Viereck, das ganz geschäftig blinkt:

■

„READY“ (gesprochen *”reddi“*) ist englisch und heißt „fertig“. Dein Computer meldet dir damit, daß er im Augenblick nichts anderes zu tun hat und bereit ist, von dir eine „Arbeit“ zu bekommen.

Das blinkende kleine Kästchen ist der *Cursor.* Ausgesprochen wird dieses englische Wort „Körser“. Dieser Cursor ist unser „Zeiger“. Er teilt uns mit, wo wir auf dem Bildschirm gerade sind. Wenn der Computer etwas tun soll, dann zeigt er es genau da, wo der Cursor sich befindet. Und wenn wir ihm etwas mitteilen oder „befehlen“ wollen, taucht der Befehl genau dort auf, wo der Cursor gerade ist.

Wie es sich für einen Zeiger gehört, können wir unseren Cursor natürlich auch bewegen.

Am vorderen Ende der Tastatur, direkt vor dir, siehst du eine ganz lange, balkenförmige Taste. Man nennt sie die „Leertaste“. Wie bei einer Schreibmaschine kannst du mit ihr einen „leeren“ Zwischenraum zwischen die einzelnen Worte beim Schreiben setzen.

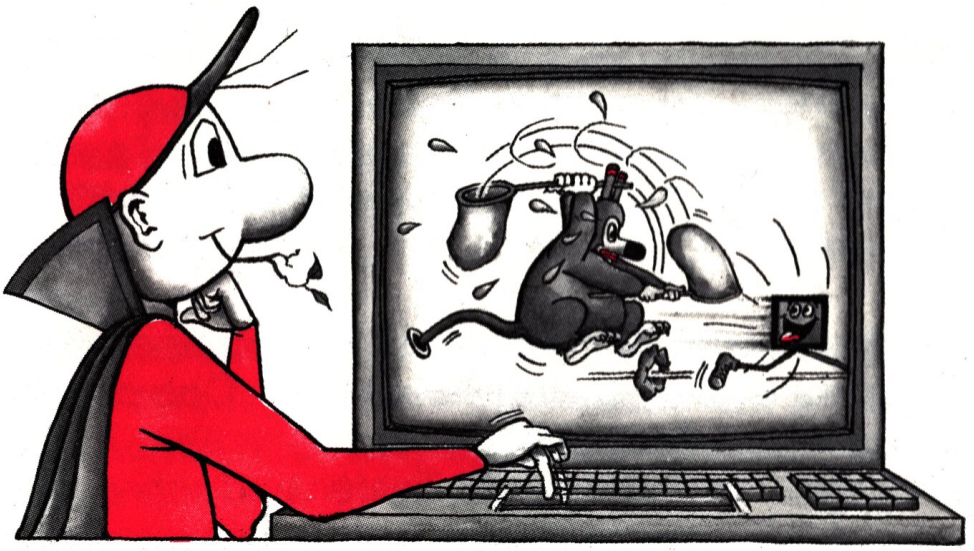
Wenn du nun *einmal* auf diese Taste drückst, bewegt sich der Cursor um *eine* Stelle nach rechts. Und beim nächsten Drücken geht er noch eine und dann noch eine. Und so weiter.

Was passiert, wenn er an den Rand vom Bildschirm kommt und du noch mal die Leertaste betätigst? Probier es aus!

Aha, der Cursor springt dann eine Zeile tiefer zum linken Bildschirmrand.

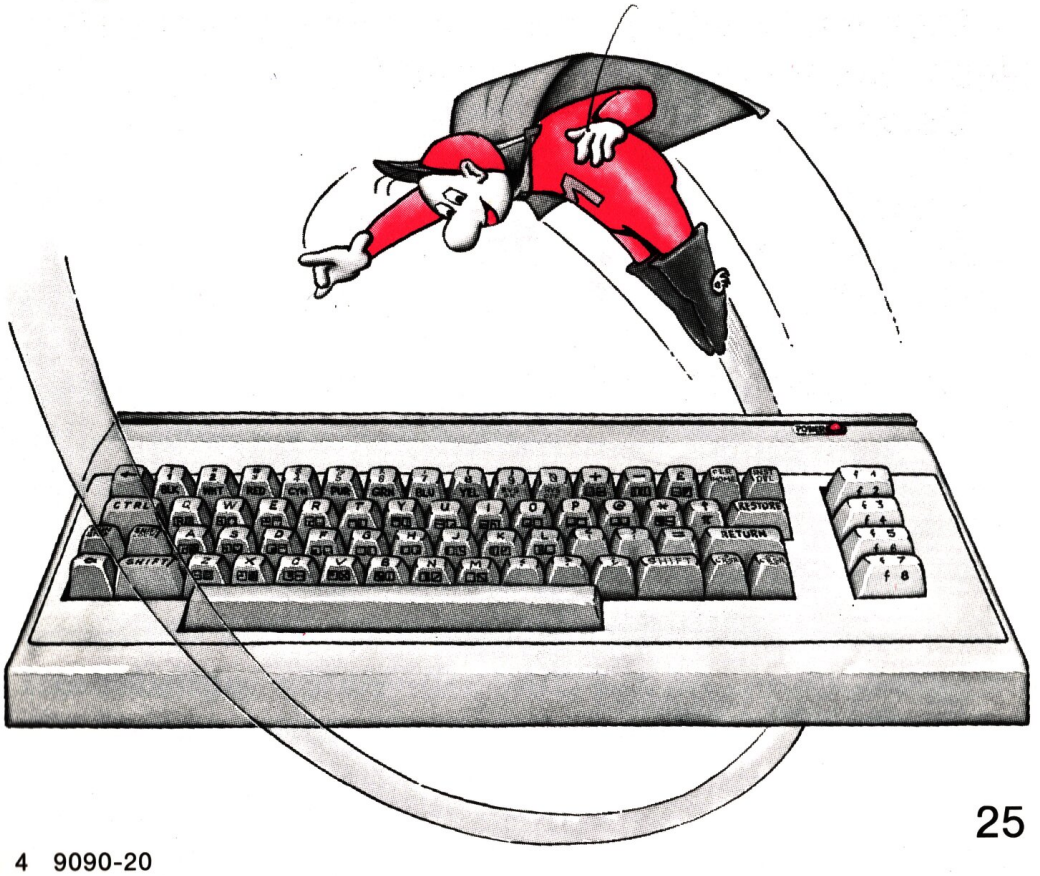
Nun betätige die Leertaste nicht einzelne Male, sondern halte sie heruntergedrückt.

Jetzt läuft er ganz schön los, nicht wahr? Cursor kann man auch als „Läufer“ übersetzen, und jetzt wissen wir auch, warum.



Um unseren Cursor zu bewegen, gibt es aber noch andere, raffiniertere Methoden. Um die herauszufinden, müssen wir uns allerdings die Tastatur unseres Computers noch etwas näher anschauen.

Du hast ja gleich gemerkt, daß das Ganze Ähnlichkeit mit einer Schreibmaschine hat. Neben einer Menge Zeichen, die es auf einer Schreibmaschine



nicht gibt, findest du auf der Computertastatur alle Buchstaben, die du zum Schreiben brauchst. Das unterscheidet einen Computer von einem Taschenrechner (obwohl beide miteinander „verwandt" sind, und man den Computer auch „Rechner" nennt.

Statt wie eine Schreibmaschine auf einem Stück Papier zu schreiben, schreibt unser Computer jedoch auf dem Bildschirm.

Jetzt hau mal kräftig in die Tasten und schreib, was du möchtest. Und vielleicht ist dir schon was aufgefallen. Versuch einmal, diesen Satz zu schreiben:

DAS ÄFFCHEN ÜBT ÖFTER ÜBERSCHLAG

Das geht also nicht. Unser Computer kann offensichtlich keine „Umlaute" schreiben. (Du weißt schon, das sind die Selbstlaute mit den zwei Pünktchen drüber – Ä, Ö und Ü.)

Das macht weiter nichts. Wir machen es ganz einfach wie bei manchen Kreuzworträtseln.

Da wird

Ä zu AE
Ö zu OE und
Ü zu UE.

Im Augenblick will der Computer ausnahmslos alles groß schreiben. Nicht alle, aber manche Computer können auch klein schreiben. Dazu gehören auch die Commodore-Computer.

Man muß den Commodore nur umschalten. Drükke einmal die beiden Tasten ganz links unten,

C = und SHIFT

gleichzeitig. Aus den Großbuchstaben sind jetzt kleine geworden.

Was passiert, wenn wir das Wort

heiß

schreiben wollen?

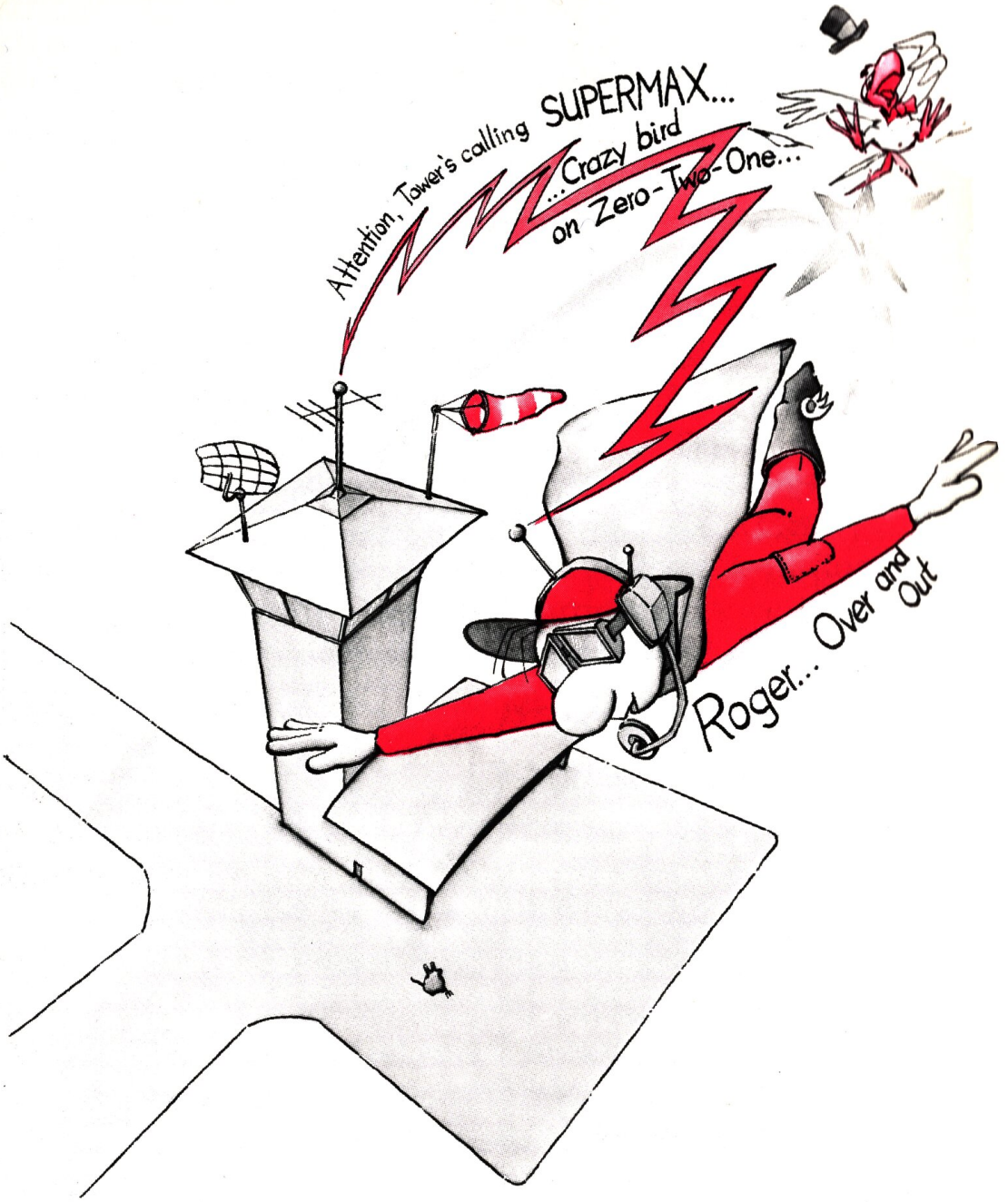
Das scharfe ß kann unser Computer auch nicht schreiben, jedenfalls vorerst nicht. Wir werden ihm aber schon noch den einen oder anderen Trick beibringen.

Jetzt sagst du dir vielleicht: Der kann ja gar nicht richtig Deutsch! Stimmt, Computer sind bis auf wenige Ausnahmen darauf getrimmt, Englisch zu verstehen und auch Englisch zu schreiben. Und im Englischen gibt es keine Umlaute und kein scharfes ß.

Aber keine Angst, wir werden ihm schon beibringen, richtig Deutsch zu schreiben. Bestimmte Antworten und „Meldungen" wird er allerdings immer in Englisch geben. Da läßt er sich nicht beirren.

Das kommt daher, daß Englisch auf der ganzen Welt die Sprache der Techniker ist. Zum Beispiel reden alle Piloten in ihren Flugzeugen englisch miteinander, oder verständigen sich mit den Flughäfen auf englisch. Auch die Kapitäne der Ozeandampfer tauschen gegenseitig Nachrichten auf englisch aus.

Deswegen hat dein Computer eine englische Tastatur. Das heißt, es fehlen nicht nur Ä, Ö, Ü und ß. Wo bei einer deutschen Schreibmaschine die Taste mit dem Z ist, siehst du beim Computer die Taste mit dem Y und umgekehrt. Das hat man deswegen so gemacht, weil im Englischen der Buchstabe Y sehr viel häufiger verwendet wird als im Deutschen. Und die Tastenreihen oben und in der Mitte sind schnel-



ler und einfacher als die übrigen zu erreichen. Das merkst du, wenn du mal ein bißchen schneller schreiben willst.

Die Kleinschrift hast du ja eingestellt, indem du gleichzeitig auf die Tasten

C =

und

SHIFT

gedrückt hast.

Um einen Großbuchstaben zu tippen, drückst du nun einfach die Taste

SHIFT

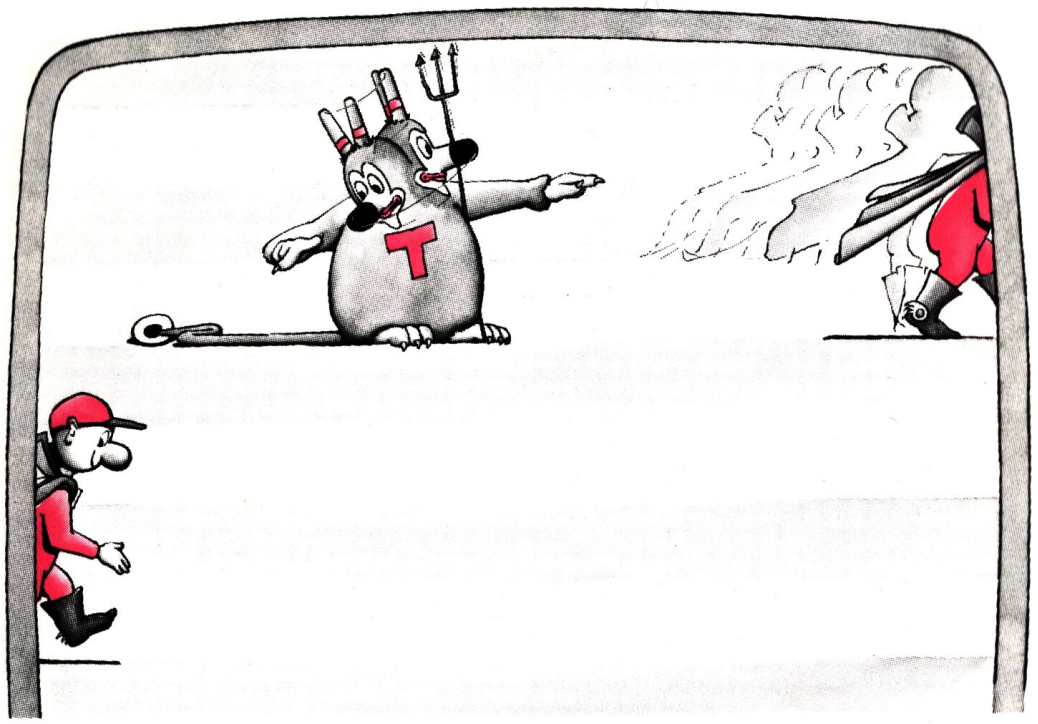
und den gewünschten Buchstaben. Weißt du noch, mit welcher Taste du deinen Zwischenraum machen kannst?

Jetzt kannst du auf dem Bildschirm schreiben wie mit einer Schreibmaschine auf dem Papier.

Mittlerweile ist der Bildschirm wohl fast vollgeschrieben. Und der Cursor bewegt sich ganz unten am Bildschirmrand. Damit du da trotzdem weiterschreiben kannst, verschiebt er nun die Zeilen auf dem Bildschirm nach oben.

Beim Schreiben ist es genauso wie vorhin bei unserem Spiel mit der Leertaste: Wenn eine Zeile voll ist, geht's einfach in der nächsten weiter. Von der Silbentrennung hat unser Computer leider überhaupt keine Ahnung. Er schreibt immer weiter. Er warnt uns auch nicht mit einem Klingelzeichen vor dem Zeilenende, wie das eine Schreibmaschine täte.

Natürlich kannst du selber auf richtiges Trennen achten. Aber wenn wir erst einmal beim Programmieren sind, wirst du schon merken, daß sich die Texte oft verschieben. Deshalb ist es zunächst besser, du gewöhnst dich daran, Computer-Texte über die Zeilenenden hinweg weiterzuschreiben und zu lesen – ohne auf Trennungen zu achten. Du wirst schon sehen; bald geht es wie von selbst.



Was meint der Max dazu? Dasselbe, wie schon auf Seite 23, und was der Max da sagt, sieht auf dem Bildschirm beim Commodore 64 so aus:

```
ALSO ICH MOECHTE, DASS WIR ENDLICH ANFANGEN!
```

und beim VC 20 so:

```
ALSO, ICH MOECHTE, DASS WIR ENDL
ICH ANFANGEN!
```

Zähle nach, wie viele Buchstaben, genauer gesagt Zeichen bei deinem Computer in eine Zeile passen. Die Zwischenräume, die man mit der Leertaste gesetzt hat, mußt du als Zeichen mitzählen.

Richtig; beim Commodore 64 passen 40 Zeichen in eine Zeile und beim VC 20 passen 22 Zeichen in eine Zeile.

Wenn du einen anderen Computer hast, zählst du die Buchstaben und Leerstellen in einer vorge-

schriebenen Zeile; dann weißt du, wieviel Zeichen bei deinem Computertyp in eine Zeile passen.

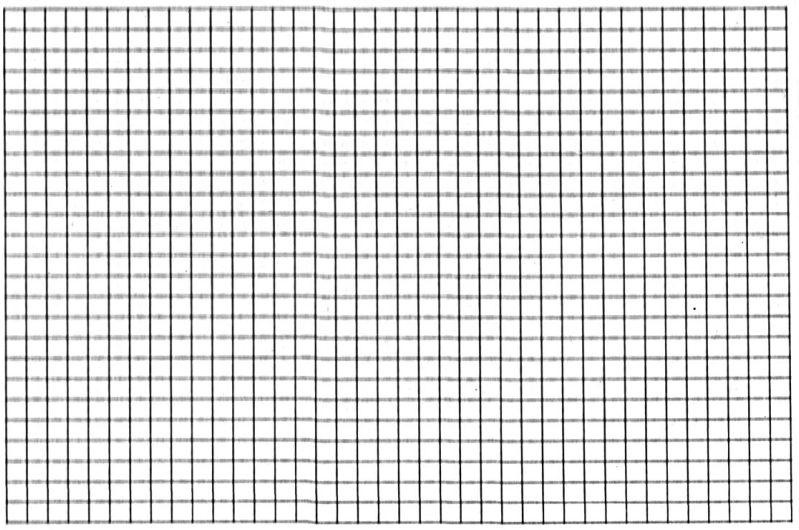
Also: ein und derselbe Text erscheint auf verschiedenen Computern in unterschiedlich langen Zeilen. Doch auch daran hat man sich schnell gewöhnt. Und daß er für die Silbentrennung noch zu dumm ist, wollen wir unserem Computerchen vorerst großzügig verzeihen.

Wie viele Zeilen passen auf deinen Bildschirm? Zähl nach:

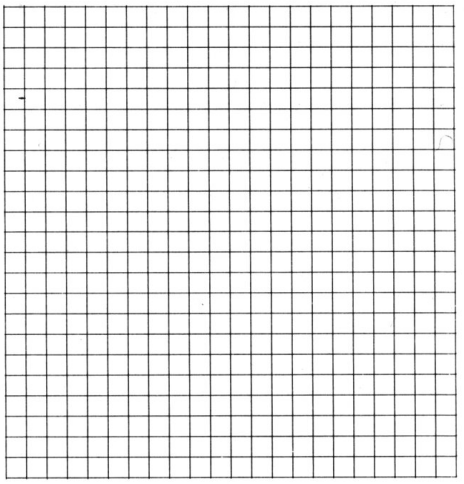
Beim Commodore 64 sind es 25 Zeilen.
Beim VC 20 sind es 23 Zeilen.
Und beim Sinclair sind es 24 Zeilen.

Also ist auch die Anzahl der Zeilen, die auf den Bildschirm passen, von Computer zu Computer verschieden.

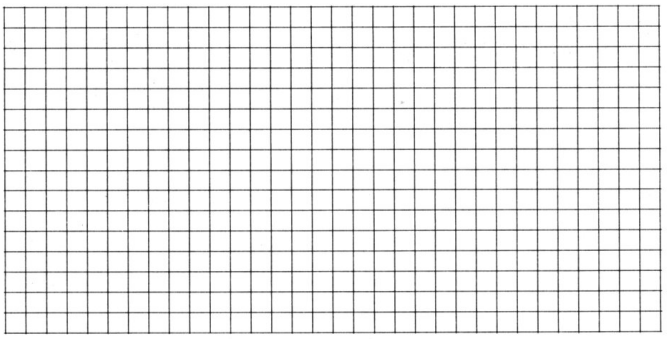
Ein Computer-Bildschirm ist aufgeteilt, wie die Kästchenseiten in einem Rechenheft, wobei das Kästchenmuster unsichtbar bleibt. In jedes Kästchen paßt ein Buchstabe oder eine Zahl oder ein anderes Zeichen. Wäre das Kästchenmuster des Bildschirms sichtbar, würde es beim C 64 so aussehen:



Das ist die Kästchen-Version eines VC 20-Bildschirms:



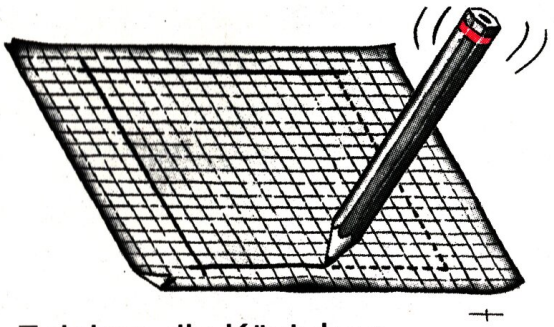
Und das ist die Kästchen-Version eines Sinclair-Bildschirms.



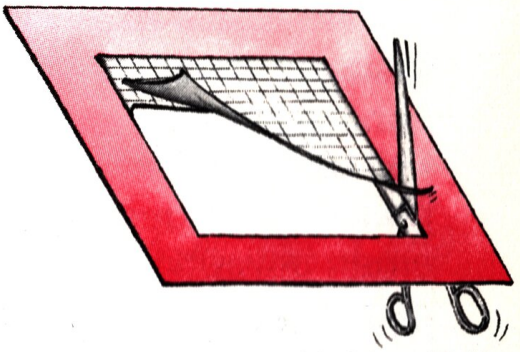
Zeichne dir die Kästchen-Version deines Computers auf kariertes Papier. Jetzt kannst du eintragen, was du später auf den Bildschirm bringen willst. So fällt es dir leicht, auch komplizierte Bilder herzustellen.

Du kannst dir eine Schablone für deinen Computer basteln, die sich immer wieder verwenden läßt:

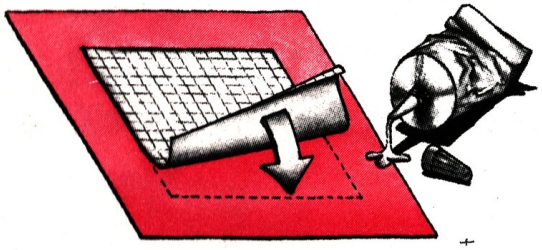
Zeichne in deine Kästchenversion, was auf den Bildschirm soll. Für jedes leere Kästchen drückst du einmal die Leertaste, für jedes volle tippst du ein „X“ ein.



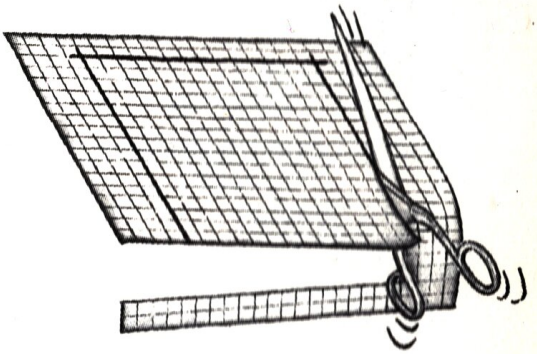
Zeichne die Kästchen-version deines Bildschirms auf Rechenpapier.



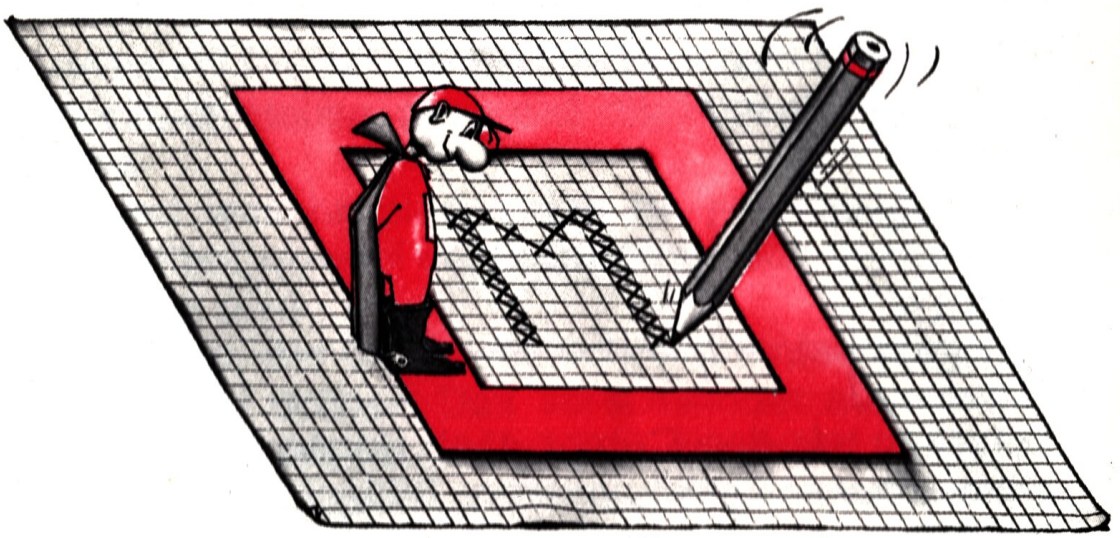
Schneide den beklebten Teil des Tonpapiers heraus.



Klebe die Kästchenversion auf stärkeres (Ton-)Papier.



Schneide die Kästchenversion deines Bildschirms aus.



Lege den so entstandenen Rahmen auf Rechenpapier, wenn du eine Figur zeichnen willst.

Halt! Bevor du mit deinen Computer-Zeichnungen loslegst, mußt du noch den vollgeschriebenen Bildschirm freimachen. Dafür gibt es mehrere Möglichkeiten. Du kannst zum Beispiel den Computer für ein paar Sekunden ausschalten. Diese Methode hat einen gewaltigen Nachteil: Wenn ein Computer ausgeschaltet wird, vergißt er alles. Er verliert sein Gedächtnis. Im Augenblick macht es nichts aus, aber später beim Programmieren wäre das sehr ärgerlich.

Besser ist, du drückst die Taste

oben rechts. Wenn man diese zugleich mit der Taste

drückt, passiert genau, was du brauchst: Der Bildschirm wird frei, und der Cursor ist wieder oben.

CLR ist die Abkürzung des englischen Wortes „clear", das heißt „klar machen", „reinigen", „säubern". Und „home" bedeutet „nach Hause" oder „heim". Mit

und zugleich

schicken wir den Cursor nach Hause (das ist die linke obere Ecke unseres Bildschirms). Dabei löschen oder „säubern" wir den Bildschirm wie eine Schultafel.

Was passiert eigentlich, wenn du die CLR/HOME-Taste ohne

SHIFT

drückst? Probier es aus! Richtig, jetzt flitzt der Cursor nach Hause, ohne daß der Bildschirm gelöscht wird.

Und was passiert, wenn du jetzt irgend etwas schreibst, obwohl der Bildschirm noch ganz voll ist? Probier mal!

Genau, der Computer schreibt einfach über das Vorherige drüber. Das verschwindet automatisch, sobald du ein neues Zeichen draufsetzt.

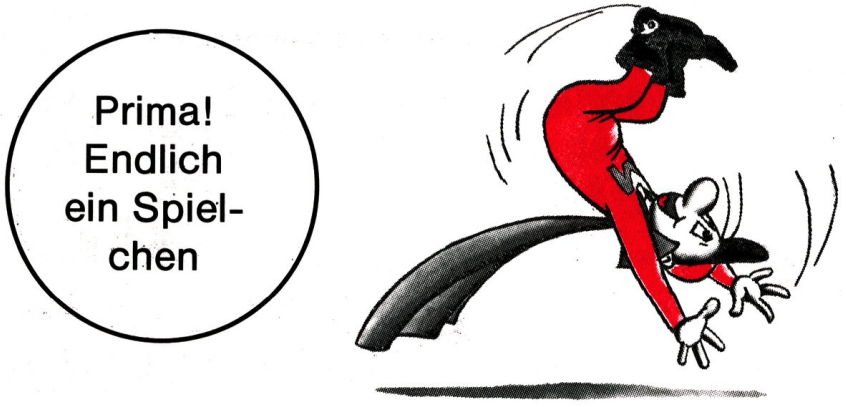
Und was passiert, wenn du ein- oder mehrmals die Leertaste drückst, und an dem Platz, wo der Cursor ist, steht schon etwas?

Aha, das Zeichen an dieser Stelle wird ersatzlos gelöscht, denn der Computer setzt nun einfach eine „Leerstelle“ darüber.

Falls wir verbessern wollen, was auf dem Bildschirm steht, haben wir nun endlich ein einfaches Mittel: Wir schicken den Cursor „heim“. Dann überschreiben wir das Alte. Wo es richtig ist, schreiben wir dasselbe noch mal, und dort, wo es falsch ist oder uns nicht gefällt, schreiben wir die Verbesserung drüber.

Mit der Zeit bekommt man dafür ein sicheres Gefühl.

Wie wäre es zwischendurch mit einem kleinen Spiel? Dabei machen wir den Cursor durch die niedergedrückte Leertaste wieder zum echten „Läufer“.



Es gibt Tasten, die man niedergedrückt lassen kann, und der Computer führt dieselbe Anweisung dann immer wieder aus. Das tut er so lange, bis wir die Taste wieder loslassen. So etwas nennt man eine „Dauerfunktion“. Bei den Commodore-Computern haben die normalen Buchstabentasten keine solche Dauerfunktion.

Hast du’s ausprobiert?

Nun also zum Spiel:

Schreibe ein Wort irgendwo in die Zeile, in der du gerade bist. Versuche nun, mit dem Cursor genau unter den ersten Buchstaben zu kommen. Am Anfang sind kurz vor dem Wort noch Einzelschritte mit der Leertaste erlaubt, aber sobald du ein bißchen Übung hast, solltest du es mit heruntergedrückter Leertaste probieren, also per Dauerfunktion. Dann tippe darunter das nächste Wort, und so weiter.
So zum Beispiel:

HALLO
(Dein Name)
HALLO

(Dein Name)
HALLO
(Dein Name)
HALLO

Na ja, immer kann es natürlich nicht klappen.

Wenn Computer-Max das gemacht hat, steht auf dem Bildschirm:

HALLO
MAX
HALLO
MAX
HALLO

Das letzte HALLO ging etwas daneben. Da hat wohl der Fehlerteufel den Monitor geschubst.

Ein anderes kleines Spiel mit dem Cursor geht so: Schreibe irgendwo in die Mitte des Bildschirms eine „aufsteigende" Zahlenreihe von eins bis fünf

12345

und dahinter

4321

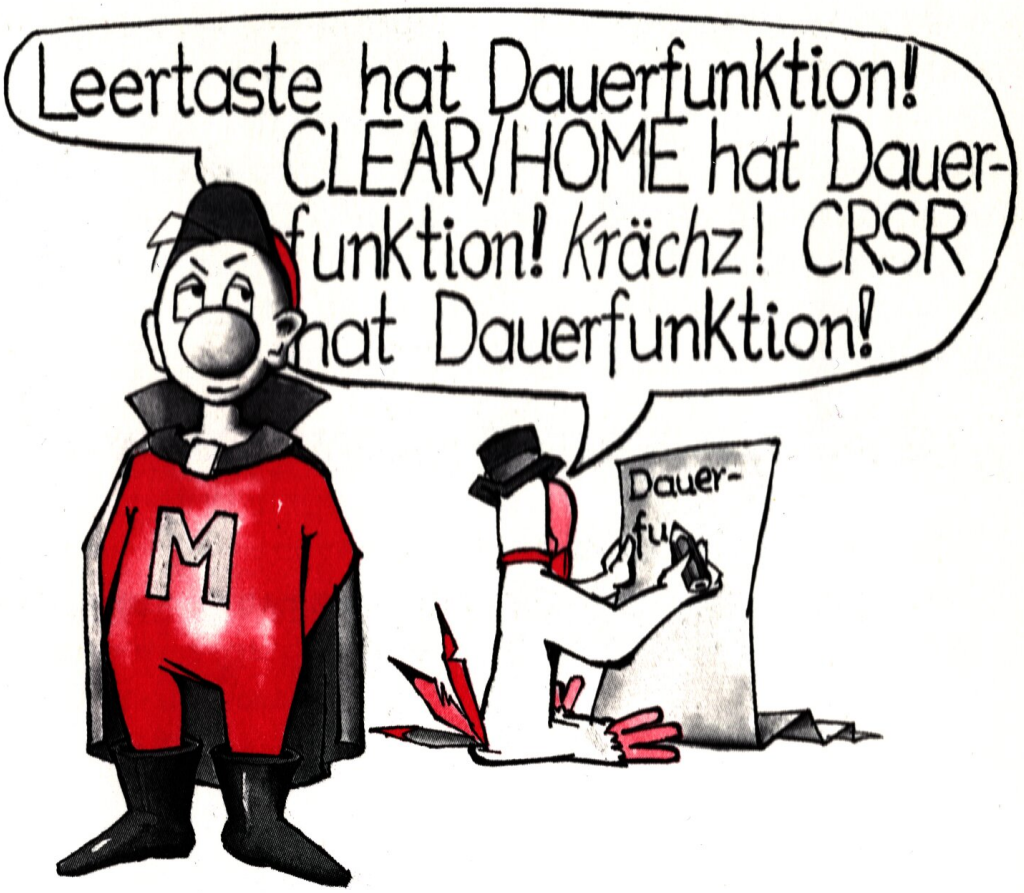
das heißt, eine „absteigende" Reihe von vier bis eins. – Solche aufsteigenden oder absteigenden Zahlenreihen werden uns später beim Programmieren noch oft begegnen.

Auf dem Bildschirm steht jetzt:

123454321

Und nun gilt es, den Cursor mit einem beherzten Druck auf die Leertaste möglichst unter die Fünf zu bekommen, und zwar *ohne Unterbrechung* – also mit *einem* Druck. Die Leertaste hat eine Besonderheit. Wie hieß das noch mal?

Die Zahl, unter die du den Cursor mit einem Druck auf die Leertaste bekommst, wird dir gutgeschrie-



ben. Landet der Cursor unter der Fünf, erhältst du 5 Punkte. Erreichst du keine Zahl, gibt's natürlich keine Punkte. Das ist ein Spiel für zwei Leute – wer wird der Champion?

Du merkst: Bei jedem Durchgang läuft der Cursor eine Zeile tiefer über den Schirm und entfernt sich so von unserer Zahlenreihe. Nach fünf oder sechs Versuchen fällt es schwer, genau zu sehen, unter welcher Zahl der Cursor eigentlich steht. Dann schreibt man sich eine neue Zahlenreihe.

Irgendwann ist der Bildschirm voll. Vielleicht drückst auch du mal versehentlich auf eine falsche Taste, und der Computer äußert sich ziemlich unver-

ständlich oder schreibt „Nachrichten“, die du (noch) nicht verstehst. Was du dagegen tun kannst, weißt du ja mittlerweile: Richtig, du löschst den Bildschirm und schickst den Cursor an den Anfang zurück, indem du die Tasten CLR/HOME und SHIFT betätigst.

Man kann das Spiel von neuem beginnen. Oder wir entwerfen zur Abwechslung Muster und Zeichnungen auf dem Bildschirm, etwa so:

XOXOXOXOXOXOXOXOXOXOXOXOXOXOXXOXO

oder so:

WMWMWMWMWMWMWMWMWMWMWMWMWMW

oder so:

=o=o=o=o=o=o=o=o

Das könnte eine Schlange sein:

==========c

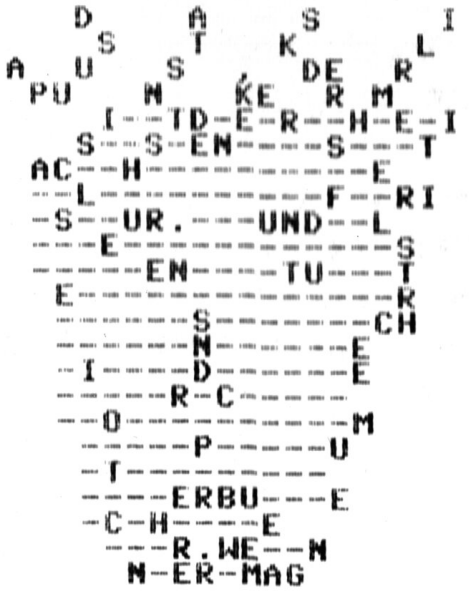
Wie wär's mit einer Treppe?

```
x
xxxx
xxxxxxx
xxxxxxxxxx
xxxxxxxxxxxxx
xxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxx
```

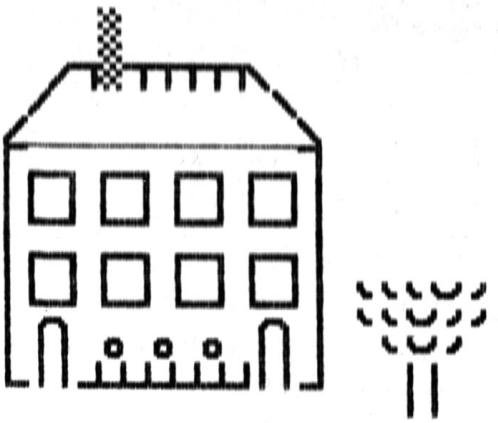
Und so weiter.

Mit ein bißchen Übung kann man die tollsten Zeichnungen aus lauter Buchstaben, Zahlen und Zeichen machen.

Ein Gesicht zum Beispiel:

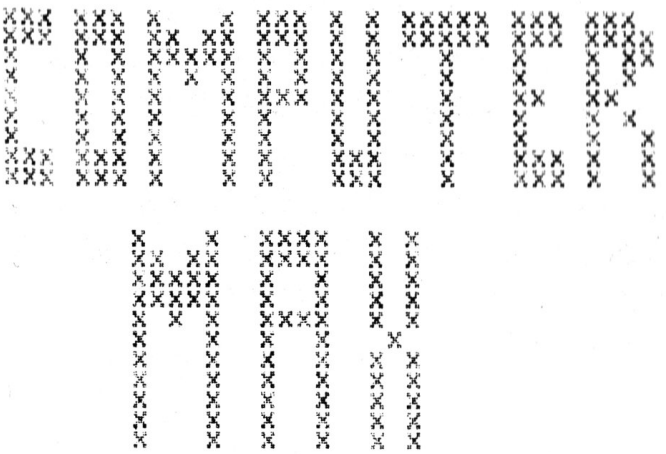


Ein Gesicht,
das man lesen kann.



Hier ist ein Haus (mit Kamin)

Und der Max hat sogar seinen Namen aus lauter Xen auf den Bildschirm geschrieben:



Vielleicht versuchst du, so etwas mit dem Anfangsbuchstaben deines Vornamens zu machen.

Dir fallen bestimmt noch viele Sachen ein, die du mit deinem Computer auf den Bildschirm bringen kannst.

Doch bei den schwierigen Bildern, „Riesen-Buchstaben“ und Wörtern macht man schnell einen Fehler. Dann ist es ziemlich mühsam, den Cursor heimzuschicken und alles noch einmal zu tippen, nur weil ein paar Fehler überschrieben werden müssen.

Also machen wir es uns etwas einfacher. Dazu haben wir nämlich die „Cursorsteuerung“. Damit läßt sich der Cursor ganz schnell an die Stelle auf dem Bildschirm bringen, an der man ihn braucht.

Hierfür verwendest du die beiden Tasten ganz unten rechts mit der Aufschrift

Das ist die Abkürzung von „Cursor“.

Bei der ersten, der linken, zeigen die Pfeile nach oben und nach unten, bei der zweiten zeigen sie nach links und nach rechts. Vielleicht denkst du dir schon, was das zu bedeuten hat.

Was tut der Cursor, wenn du auf die erste der beiden Tasten drückst? Probier's doch mal!

Ja, er springt nach unten.

Und was bewirkt die zweite Taste?

Richtig, der Cursor läuft nach rechts!

Wenn wir den Cursor nach oben oder nach links bewegen wollen, hilft uns die „Umschalttaste“

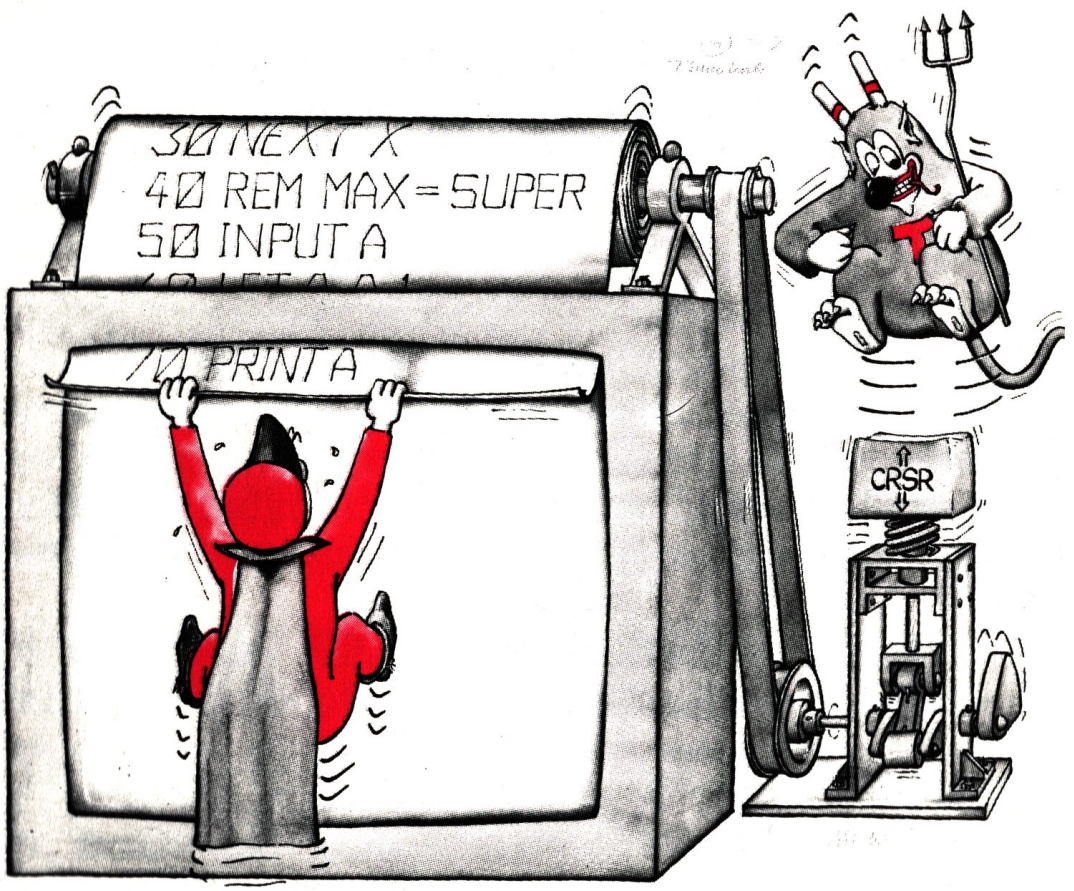
Die haben wir schon beim Groß- und Kleinschreiben kennengelernt. Die erste Cursor-Taste zusammen mit SHIFT gedrückt, und der Cursor springt

nach oben. Die zweite Cursor-Taste, zusammen mit SHIFT gedrückt, läßt ihn den Rückzug nach links antreten.

Jetzt solltest du etwas üben, um mit dieser „Cursorsteuerung“ zurechtzukommen. Was passiert zum Beispiel, wenn der Cursor an die Ränder des Bildschirms rechts oder links, unten oder oben anstößt? Du kannst es dir nicht denken? Da gibt's nur eins – Probieren!

Falls der Cursor, am unteren Bildschirmrand angekommen, immer weiter nach „unten“ geht, schiebt er die oberen Zeilen aus dem Bildschirm.

Bis jetzt wissen wir noch nicht, wie wir diese Zeilen wieder zurückholen. So lange müssen wir also vorsichtig sein.



Ansonsten können wir den Cursor auf dem Bildschirm steuern, wohin wir wollen.

Mit der Zeit bekommst du für das Herumwandern auf dem Bildschirm das richtige Gefühl, und du bewegst deinen Cursor so gekonnt wie ein Rennfahrer seine Formel-I-Maschine.

Um das noch ein wenig zu üben, haben wir zwei Spiele entwickelt.

Falls du das Buch zusammen mit einer Diskette gekauft hast, kannst du diese Spiele gleich von der Diskette in den Computer laden. Sonst mußt du entweder selber das Programm eintippen oder dir von jemand, der schon gut tippen kann, dabei helfen lassen. Das Programm steht im Anhang auf den Seiten x bis x.

Was du auf diesen Seiten siehst, nennt man ein LISTING. Beim Eintippen eines solchen Listings darf man keine Fehler machen. Schon ein vergessener Punkt oder ein falscher Zwischenraum, und das Programm läuft nicht. Stattdessen bekommst du höchstwahrscheinlich eine Fehlermeldung. Also

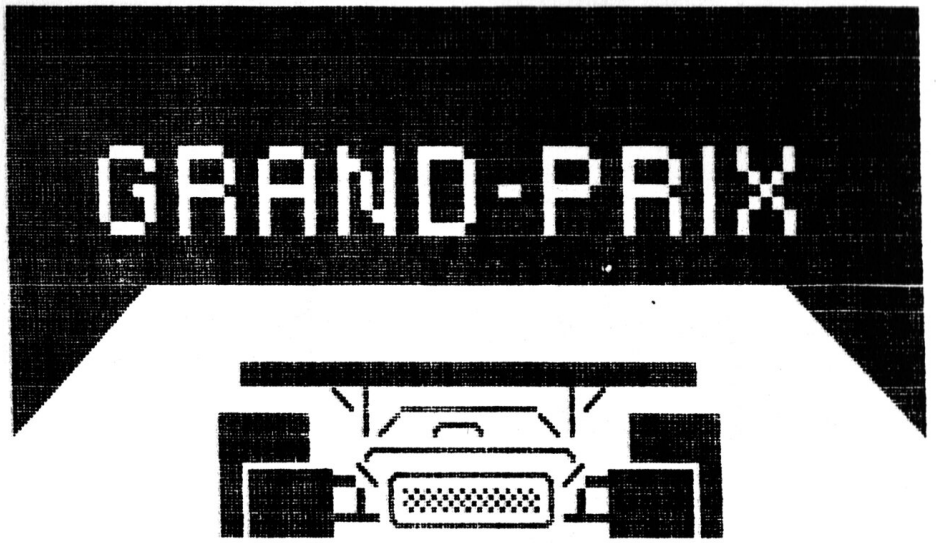
paß gut auf beim Eintippen, und lies das Programm dann zur Kontrolle noch mal durch – am besten mit jemand anderem zusammen, denn vier Augen sehen mehr als zwei!

Das erste Spiel heißt **„SCHLANGENSPIEL“.** Der Computer zeichnet dir dabei eine Folge von „X“en und Strichen auf den Bildschirm. Die sollst du genau nachzeichnen, in einem Zug und ohne die Leertaste als „Radiergummi“ zu Hilfe zu nehmen. Wenn du es richtig gemacht hast, erscheint ein alter Freund auf dem Bildschirm.

```
WIE HEISST DU? MAX

WOLLEN WIR EIN KLEINES SPIEL MACHEN,
MAX?
GEBE >JA< ODER >NEIN< EIN!
?

ICH SCHREIBE EINE REIHE VON >X< UND >-<
AUF DEN BILDSCHIRM
ZUM BEISPIEL SO:
-X-XX---XXX-XX-X-X
UND DU KANNST DANN DIESE REIHE NACHTIP-
PEN, MAX

BITTE GIB MIR NUN EINE ZAHL ZWISCHEN
1 UND 20 EIN.
JE GROESSER DIE ZAHL IST, DESTO SCHWIE-
RIGER WIRD'S
?
```

Das 2. Spiel heißt **GRAND-PRIX-SPIEL** und ist ein rasantes Rennen:

Du bist ein Formel-I-Pilot, der seinen Rennwagen sicher über eine schwierige Rennstrecke lenken muß.

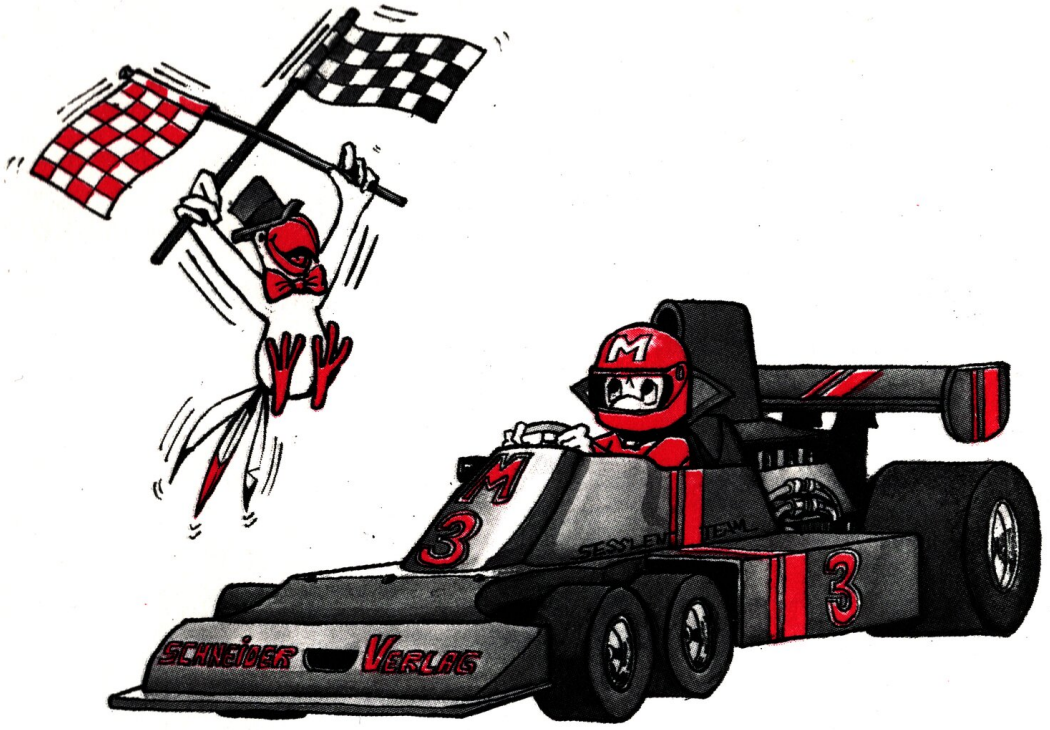


Hallo!!
Du also willst Formel-1-Fahrer werden?
Dazu mußt du mir noch ein paar Fragen richtig beantworten.
Dein Name? Max

Natürlich verwendest du dazu die Cursorsteuerung. Zuvor mußt du aber noch ein paar Fragen beantworten, damit du deine Fahrerlizenz erhältst.

Der Computer mißt dabei die Zeit, und wenn du gut durchgekommen bist, steht dein Name schließlich auf der Siegerliste.

Wenn du falsch steuerst, gibt es einen Unfall – einen „Crash“ (sprich „Kräsch“) –, wie es in der Sprache der Rennfahrer (und Computerfachleute) heißt. Nur gut, daß auf unserem Bildschirm alle Unfälle glimpflich verlaufen. Dann gilt es eben, eine neue Runde zu wagen.



Sofern dein Spiel schon auf einer Kassette oder einer Diskette gespeichert ist, mußt du es zuerst in den Computer *laden.* Dazu muß der Cursor ganz links am Zeilenanfang stehen. Um ein Programm von der Kassette in den Computer zu schleusen, benützt man einen Kassettenrecorder. Nachdem du die Kassette in den „Datenrecorder“ gelegt hast, tippst du

LOAD ”SCHLANGENSPIEL“

oder

LOAD "GRAND PRIX"

LOAD (sprich "lohd") ist ein englisches Wort und heißt „laden". „Laden" ist Computer-Chinesisch und bedeutet, ein Programm in den Computer zu packen – etwa das **GRAND-PRIX-SPIEL.** Vor dem Laden ist das Programm auf einem Datenträger. Dieser Datenträger kann eine Kassette oder eine Diskette sein.

Um ein Programm von der Diskette in den Computer zu laden, brauchst du ein Diskettenlaufwerk, auch Diskettenstation genannt. Wie man die Diskette in ein solches Diskettenlaufwerk einlegt, steht in dem dazugehörigen Handbuch. Du kannst ja auch jemanden bitten, dir dabei zu helfen.

Beim Laden des Programmes *von der Diskette* mußt du auf deinem Commodore 64 folgendes tippen:

LOAD "SCHLANGENSPIEL", 8

oder

LOAD "GRAND PRIX", 8

Danach drückst du die Taste mit der Aufschrift

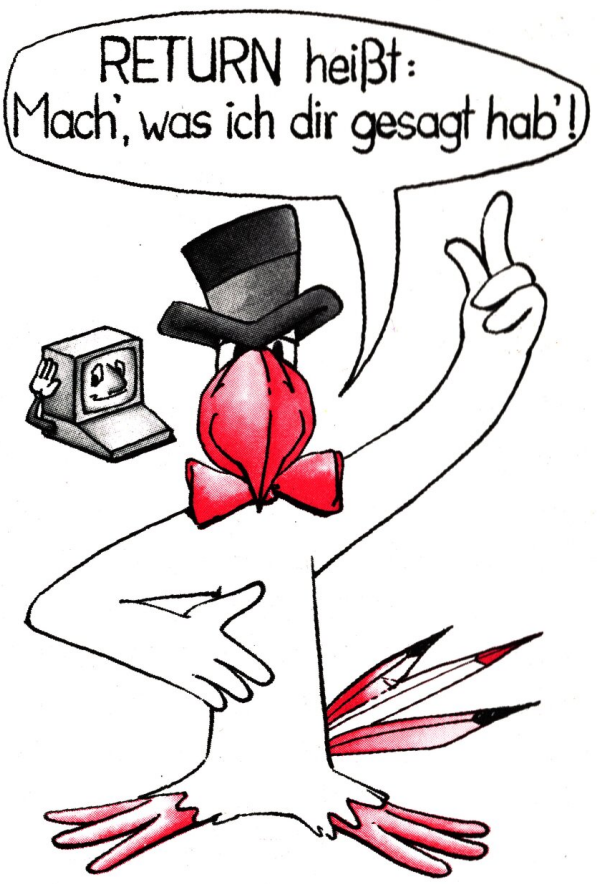
RETURN

Auf der Commodore-Tastatur findest du sie rechts außen auf halber Höhe.

Die RETURN-Taste ist überhaupt eine der wichtigsten Tasten auf dem Computer. Sie sagt ihm nämlich, daß er das, was du ihm zuvor eingegeben hast, *wirklich* ausführen soll.

„RETURN“ (sprich „ritörn“) heißt für den Computer soviel wie: Nun mach mal!

Ohne die RETURN-Taste geht in der Computerei eigentlich gar nichts.



Wenn du von einer Kassette in den Computer lädtst, antwortet der Computer auf deinen LOAD-Befehl nach dem Bestätigen der Return-Taste:

PRESS PLAY ON TAPE

Damit fordert er dich auf, die Taste mit der Aufschrift

PLAY

auf dem Kassettenrecorder zu drücken. Nun läuft das Band, und der Computer sucht darauf das Programm mit dem Namen, den du eingetippt hast.

Beim Laden von der Diskette ist das etwas anders.

Da fängt das Diskettenlaufwerk gleich nach dem Drücken der Return-Taste selbsttätig zu schnurren an. Daß er sucht, sagt dir der Computer mit der Meldung:

OK
SEARCHING FOR "SCHLANGENSPIEL"

Beim Formel-1-Spiel lautet die Meldung:

OK
SEARCHING FOR "GRAND PRIX"

Beim C 64 „verschwindet" während des Ladens der Bildschirm für eine Zeit. Währenddessen, das kann man sich gut vorstellen, sucht der Computer das gewünschte Programm.

Die Anführungszeichen, auf die man beim Suchen eines Programms nicht verzichten darf, hast du sicher gefunden. Sie sind wie bei einer Schreibmaschine auf der Taste mit der „2". Und wie bei der Schreibmaschine werden die Anführungszeichen mit Hilfe der Umschalttaste geschrieben. Die Umschalttaste auf unserem Computer heißt, das weißt du ja schon, „SHIFT".

Sobald du die Taste mit Aufschrift

zusammen mit der SHIFT-Taste drückst, kommt das Anführungszeichen auf den Bildschirm. *Wie die Schreibmaschine kennt auch der Computer nur das Anführungszeichen oben.*

Du hast jetzt den Computer richtig angewiesen, ein bestimmtes Programm oder ein bestimmtes

Spiel von der Diskette zu „laden" – indem du zum Beispiel

LOAD "GRAND PRIX", 8

getippt hast. Dann meldet er dir zuerst, daß er das Programm sucht:

OK
SEARCHING

Was OK heißt, weißt du sicher.
Dann meldet dein Computer

LOADING (Hier steht der Name des Programms)

Und schließlich hält er sich für

READY. (zu deutsch: „fertig", „bereit")

Er ist also bereit, wieder mit uns zu arbeiten, nachdem er das Programm geladen hat.

Nun soll es losgehen. Dazu mußt du noch das Programm starten. Dazu tippst du die Anweisung

RUN

Du schreibst also RUN auf den Bildschirm, und zwar am Zeilenanfang links, sonst gibt es eine Fehlermeldung.

Welche Taste mußt du nach der Anweisung RUN noch betätigen? (Dein Computer muß ja noch erfahren, daß er nun ausführen soll, was du ihm befohlen hast.)

Genau, du mußt noch die

RETURN -

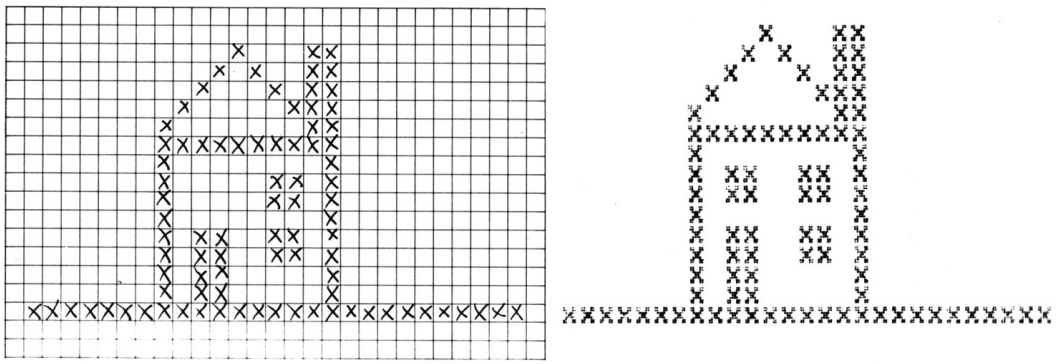
Taste drücken. Jetzt erst geht das Spiel los, das du

mit diesen Anweisungen vom „Datenträger“ in deinen Computer geladen hast.

Wenn du unsere beiden Spiele ein paarmal gespielt hast, beherrschst du sicher die Bewegungen auf deinem Computer-Bildschirm schon sehr gut. Dann kann dir so schnell keiner mehr etwas vormachen.

Mit so viel Übung können wir auch ein bißchen schwierigere Bilder auf den Bildschirm zaubern. Dazu sollten wir sie auf Rechenpapier vorzeichnen. Vorher müssen wir auf dem karierten Papier die Kästchen-Version unseres Computer-Bildschirms eintragen.

Hier ist zum Beispiel ein Haus, zuerst in der Kästchen-Version, dann als Bildschirm-Zeichnung.

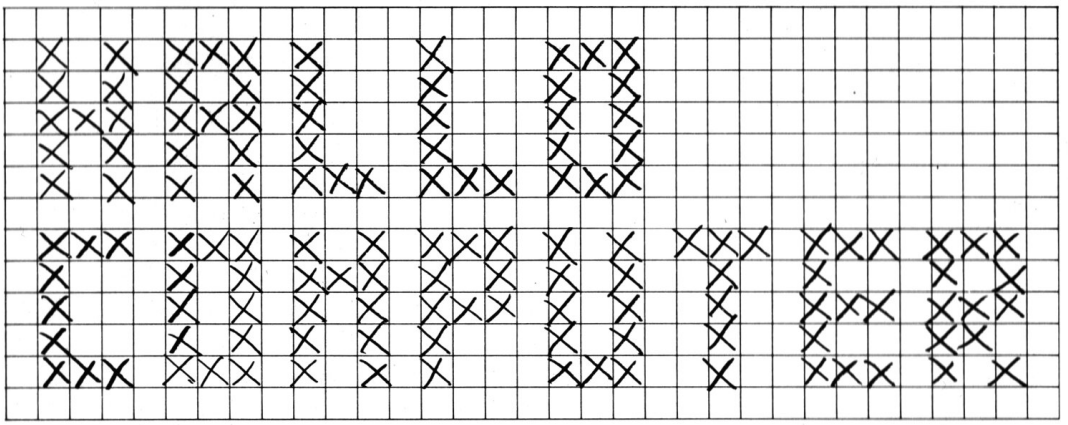


Das war die Version für den Commodore 64. Besitzt du einen anderen Computer, zeichnest du das Haus in die Kästchen-Version deines Bildschirms. Dann brauchst du sie nur noch auf den Bildschirm zu übertragen.

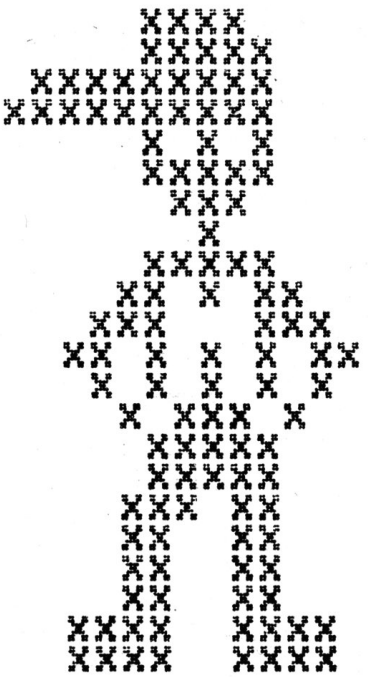
So schreibt man

HALLO COMPUTER

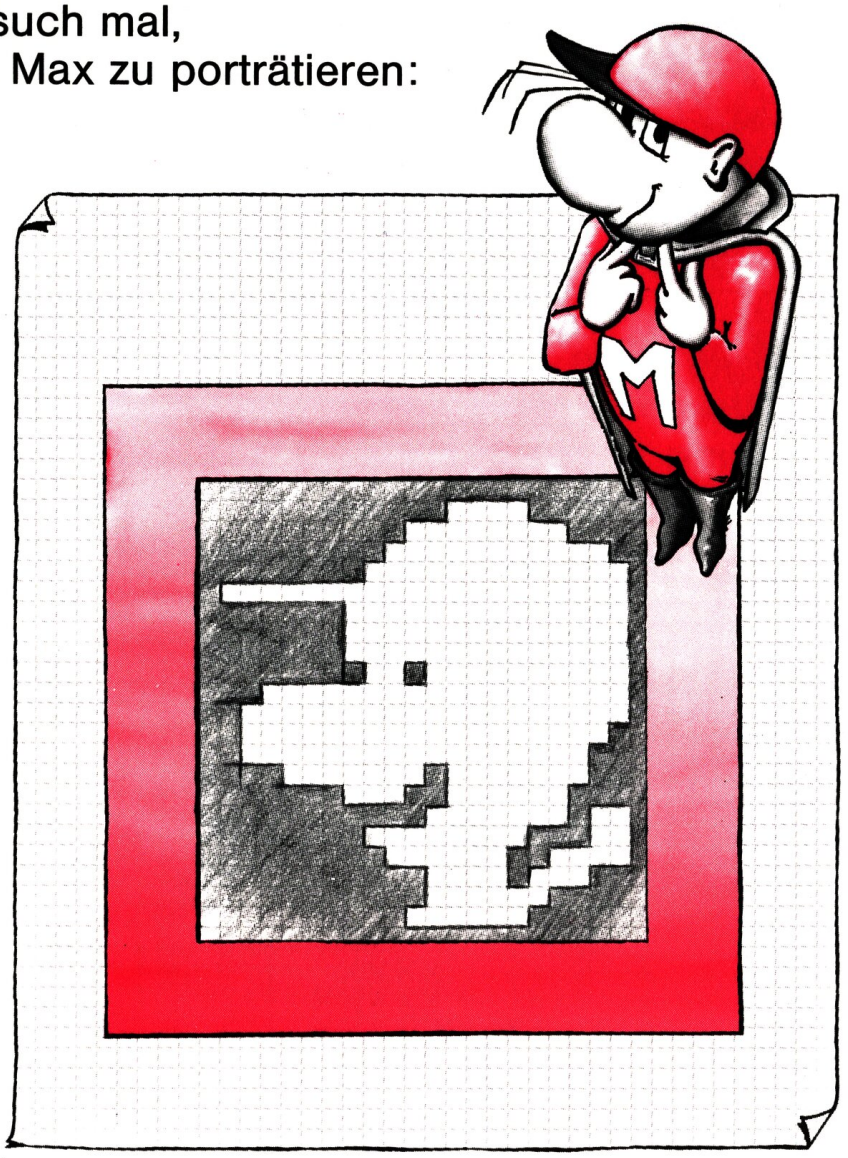
in „Riesenbuchstaben“ auf den Bildschirm:



Auch unseren Freund, den Computer-Max kannst du auf den Bildschirm bringen. Das sieht dann so aus:



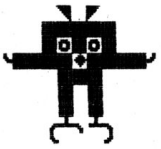
Versuch mal,
den Max zu porträtieren:

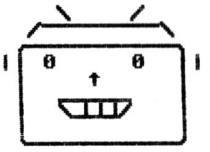


Wie du mit Hilfe der Cursorsteuerung überall hinkommen und mit der Leertaste „radieren“ oder löschen kannst, weißt du ja mittlerweile. Da macht es dir sicher nichts aus, wenn du dich einmal vertippt hast und etwas verbessern mußt.

Es wird dir sicher auch keine großen Schwierigkeiten machen, zum Beispiel eine ganze Stadt zu zeichnen – zuerst auf dem Rechenpapier, dann auf dem Bildschirm.

Wie wär's mit ein paar Monstern oder vielleicht ein paar Faschingsmasken?





Dazu verwenden wir nicht mehr bloß die „X"-Buchstaben, sondern auch andere Buchstaben und Zeichen. Trau dir ruhig zu, daß du die geeigneten Zeichen selber herausfindest. Probier einfach ein bißchen mit den Umschalttasten

C =

und

SHIFT

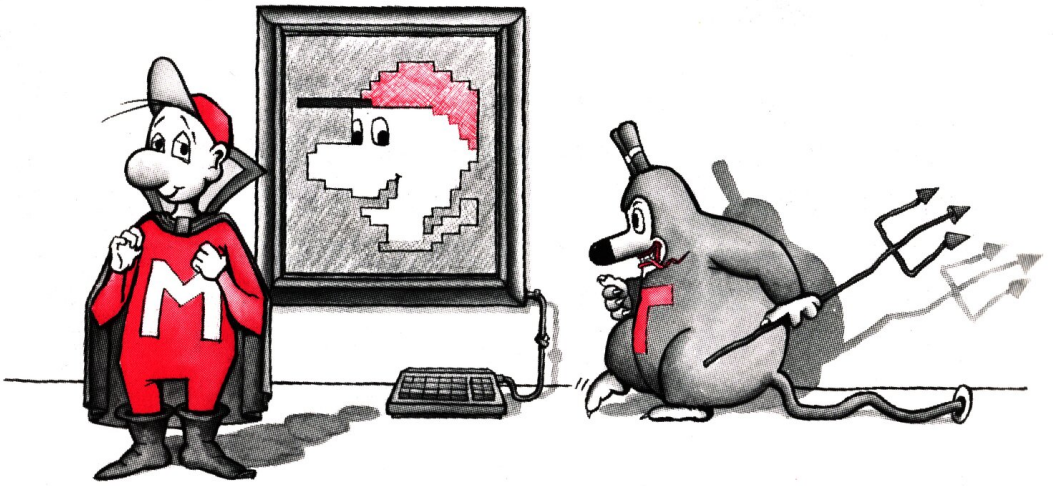
herum. Wenn etwas Falsches herauskommt, oder der Computer Fehlermeldungen schreibt, kannst du ja immer noch mit unserer Cursorsteuerung und der Leertaste als „Radiergummi" verbessern. Fehlermeldungen beginnen alle mit einem „?" und sagen, daß der Computer etwas nicht richtig verstanden hat.

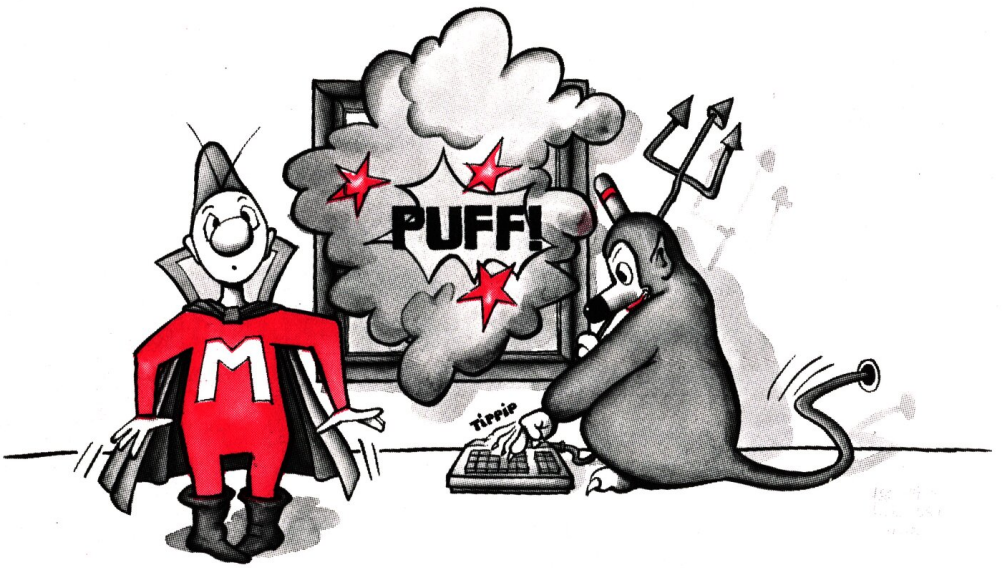
Aber bald entwirfst du ja sowieso deine eigenen Zeichnungen, und die sind möglicherweise noch viel toller als unsere Vorschläge.

Was heißt hier Print?

Es macht bestimmt Spaß, eigene Bildschirm-Kunstwerke zu schaffen – auch wenn man zwischendurch heftig „radieren“ muß.

Aber ärgerlich ist es doch, daß jede Zeile unserer Bildschirm-Zeichnung ein für allemal verlorengeht, sobald sie einmal über den oberen Rand hinausgerutscht ist.





Ein bißchen ist das wie bei uns selber. Vieles was wir hören, geht beim einen Ohr hinein und beim anderen gleich wieder hinaus. Natürlich gibt es ein Mittel dagegen: Man muß es sich fest einprägen.

Damit der Computer sich etwa eine deiner Zeichnungen länger merken kann, mußt du sie in ein Programm verpacken.

Ein Programm behält der Computer so lange im Gedächtnis, bis wir es löschen. Ein Programm kann auch abgespeichert werden. Das bedeutet, es wird auf eine Kassette oder eine Diskette übertragen. So kann es auch dann nicht mehr verlorengehen, wenn wir den Computer abschalten. Wie bei einer Schallplatte oder Musikkassette kannst du ein so gespeichertes Programm so oft „spielen“ wie du willst.

Das Wort „Programm“ kennst du wohl hauptsächlich vom Fernsehen oder vom Theater. Schau dir so ein Programm in der Tageszeitung oder auf einer Litfaßsäule an. Eine gewisse Ähnlichkeit mit solchen Programmen hat auch unser Computer-Programm.

Jedes Programm, egal ob beim Funk, Fernsehen

oder Kino, teilt uns mit, was zuerst, was als zweites, was als drittes kommt, und so weiter – vom Anfang bis zum Ende.

Beim Fernsehprogramm zum Beispiel steht vor den einzelnen Programmteilen die Uhrzeit. So weißt du, wann du einschalten mußt, um eine Sendung zu sehen, die dich interessiert. Bei einem Theaterprogramm stehen vor den einzelnen Teilen des Theaterstücks Nummern. Man hat damit einen groben Überblick darüber, was in dem Theaterstück geschehen wird, wann die Pause kommt, und so weiter.

Auch unser Computerprogramm erhält Nummern. Diese Nummern sollen uns *und dem Computer* sagen, in welcher Reihenfolge die einzelnen Pro-

grammteile erledigt werden sollen. Programmteile sind zum Beispiel:
- eine Rechenaufgabe lösen
- etwas auf den Bildschirm schreiben
- etwas auf den Bildschirm zeichnen
- einen Ton erzeugen

und so weiter.

Diese einzelnen Teile eines Computerprogramms nennt man *Programmzeilen. Und jede Programmzeile bekommt eine eigene Nummer.* Eine solche Programmzeile kann aus einer kurzen Rechnung bestehen. Sie kann aber auch länger sein als eine normale Bildschirmzeile. Dann schreibt der Computer einfach in der nächsten Zeile weiter, wie wir das von vorhin bereits kennen.

„Ewig" lang dürfen Programmzeilen aber auch nicht werden. Sonst muß der Computer zu viel auf einmal erledigen. Dann kann es vorkommen, daß er einfach streikt.

Für das Schreiben eines solchen Programms gibt es eine Reihe von Befehlen, Anweisungen, Fragen und so weiter. Sie bilden die Programmier-Sprache BASIC, von der wir schon ganz am Anfang gesprochen haben.

Zwar unterscheiden Fachleute ganz streng zwischen Befehlen, Kommandos, Anweisungen und Funktionen. Aber für uns geht es im Augenblick nur darum, dem Computer zu sagen, was er tun soll.

Als erstes soll er etwas auf den Bildschirm schreiben. Der Befehl lautet

PRINT

„Print" ist das englische Wort für „drucken" oder „schreiben".

Alles, was der Computer schreiben soll, muß nach diesem Befehl in Anführungszeichen stehen.

Zum Beispiel

PRINT „THOMAS“

Der Computer soll nun in einem Programm das Wort „Computer“ schreiben. Dazu verwenden wir den Befehl:

PRINT „COMPUTER“

Und nun bekommt unsere Programmzeile noch eine Nummer. Man geht dabei in Zehnerschritten vor, schreibt also 10, 20, 30, 40. . .

Kannst du dir denken, warum man in Zehnerschritten numeriert?

Falls wir zwischen den einzelnen Programmzeilen noch Nummern freihalten, kann man im Bedarfsfall später noch Zeilen einfügen. Der Computer „arbeitet“ die Programmzeilen nach der Größe ihrer Nummern ab.

Das probieren wir gleich einmal aus.

Tippe also bitte unsere erste Programmzeile ein:

10 PRINT „COMPUTER“

Wenn du damit fertig bist, tut sich erst einmal gar nichts. Die Zeile steht da, und dahinter blinkt der Cursor.

Um die Programmzeile ins „Gedächtnis“ des Computers zu bringen, mußt du die Taste mit der Aufschrift

RETURN

drücken. Nun weiß der Computer, daß er die Programmzeile in das Programm aufnehmen soll. Er weiß auch, daß er sie ausführen soll, wenn es mit dem Programm losgeht.

Losgehen heißt soviel wie „starten". Wie startest du das Programm? Vorläufig ist es nur gespeichert. Eins solltest du jetzt auf keinen Fall tun: den Computer ausschalten und eine Tüte Kakao holen. Das mit dem Kakao wäre ja okay. Aber wenn du ausschaltest, vergißt der Computer alles – auch unser kleines, einzeiliges Programm.

Schau auf den Bildschirm: Nach dem Drücken der Return-Taste ist der Cursor an den Anfang der nächsten Zeile gesprungen. „Return" heißt unter anderem „zurückkehren". Genau das macht der Computer: Er schleppt den eingegebenen Befehl in sein Gedächtnis und gibt dir dann den Cursor dorthin zurück, wo du weiterarbeiten kannst.

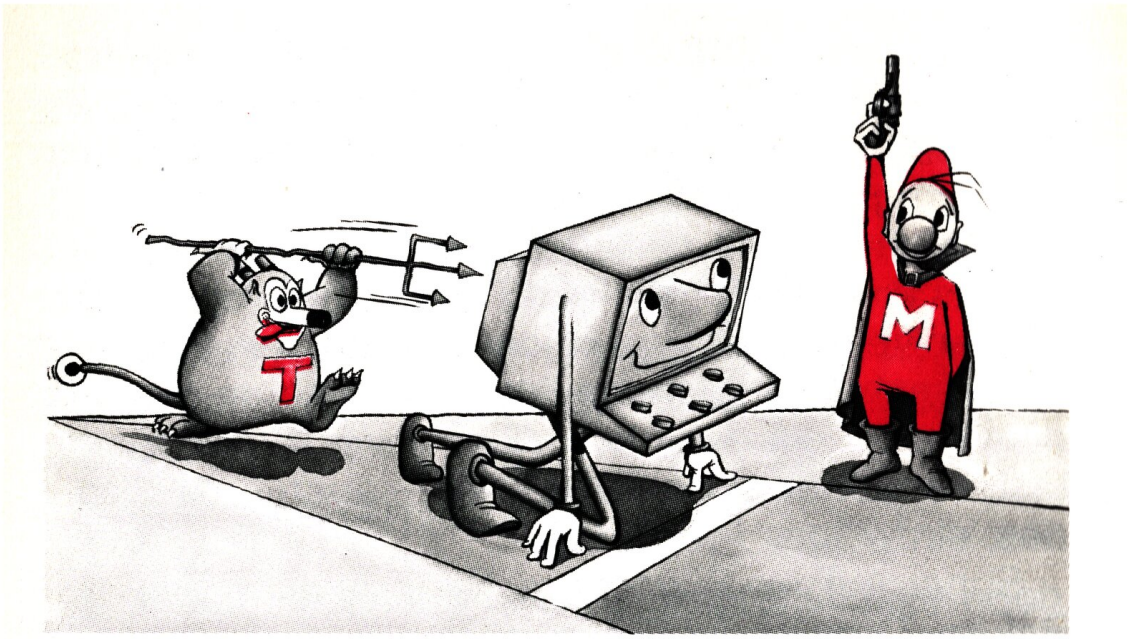
Nun hast du zwei Möglichkeiten. Entweder schreibst du das Programm weiter, oder du „startest" es. Das BASIC-Wort, mit dem ein Programm gestartet wird, heißt

RUN

„RUN" erhält keine Zeilennummer, denn es gehört nicht zum Programm. „RUN" ist vielmehr das Startkommando für das Programm.

„RUN" (sprich „ran") heißt auf englisch „laufen". Man kann sich gut vorstellen, wie der Computer für das Programm den Start freigibt.

Hier kann man sich den Unterschied zwischen Befehl und Kommando klarmachen: Was man dem Computer *innerhalb* eines Programmes aufträgt, nennt man einen „Befehl". Was man dem Computer



außerhalb eines Programmes (das heißt nach der letzten Programmzeile) aufträgt, nennt man ein „Kommando". Das mußt du dir jetzt aber nicht unbedingt merken.

Was passiert nun, wenn du das Programm

```
10 PRINT "COMPUTER"
```

mit

```
RUN
```

und danach der RETURN-Taste startest?
Auf dem Bildschirm steht zunächst noch

```
10 PRINT "COMPUTER"
RUN
```

Nachdem du „RUN" geschrieben hast, kommt auf dem Bildschirm

```
COMPUTER
READY.
```

hinzu. Sicher, das hättest du auch ohne Programm auf den Bildschirm schreiben können. Aber dieses „Mini-Programm“ hat einen Vorteil. Das merkst du, wenn du nun noch einmal

```
RUN
```

eintippst und das Programm dann durch Betätigen der RETURN-Taste wieder startest. Du siehst, jetzt erscheint das Wort „Computer“ von neuem, ohne daß du es extra noch mal schreiben mußtest. Nun stell dir vor, es hätte sich um das Wort „Donaudampfschiffahrtskapitänsleutnantsstiefelputzerlehrling“ gehandelt . . .

Wenn man also ein Programm geschrieben hat, egal ob ein ganz kurzes oder ein ganz langes, kann man es immer wieder starten.

Du kannst das Programm natürlich auch wieder löschen, das heißt, aus dem Gedächtnis des Computers entfernen.

Eine Möglichkeit, das Gedächtnis des Computers zu löschen, haben wir schon kennengelernt: Sobald du den Computer auch nur für ein paar Sekunden ausschaltest, verliert er sofort sein Gedächtnis, und du kannst ein neues Programm schreiben.

Aber es gibt einfachere Methoden, ein Programm oder einen Teil von einem Programm zu löschen.

Probieren wir es doch einfach mal mit dem „Ausradieren“. Du steuerst den Cursor an den Anfang der Zeile. Darin bist du wahrscheinlich schon ein echter „Profi“. Dann „radierst“ du mit der Leertaste. Und wenn die Zeile weg ist, versuchst du das Mini-Programm noch einmal zu starten. Also tippe

```
RUN
```

und drücke dann die RETURN-Taste. Was passiert?

Genau – das „Radieren" mit der Leertaste fegt das Programm zwar vom Bildschirm, löscht es aber nicht aus dem Gedächtnis des Computers. Obwohl du das Programm nicht mehr auf dem Bildschirm siehst, kannst du es trotzdem noch so oft laufen lassen, wie du willst.

Daß unser Programm vom Bildschirm verschwunden ist, ärgert uns eigentlich. Denn wir wissen ja jetzt nicht mehr, wie es aussieht.

Wie holen wir das Programm wieder auf den Bildschirm zurück? Es gibt ein Kommando, mit dem wir unser Programm wieder auf den Bildschirm bekommen – so, wie wir es am Anfang geschrieben haben.

Ein Kommando geben wir dem Computer nicht innerhalb eines Programms, sondern tippen es extra auf den Schirm. Das Kommando (ohne Programm-Nummer!) wird wie immer an den Anfang der Zeile geschrieben. Du tippst

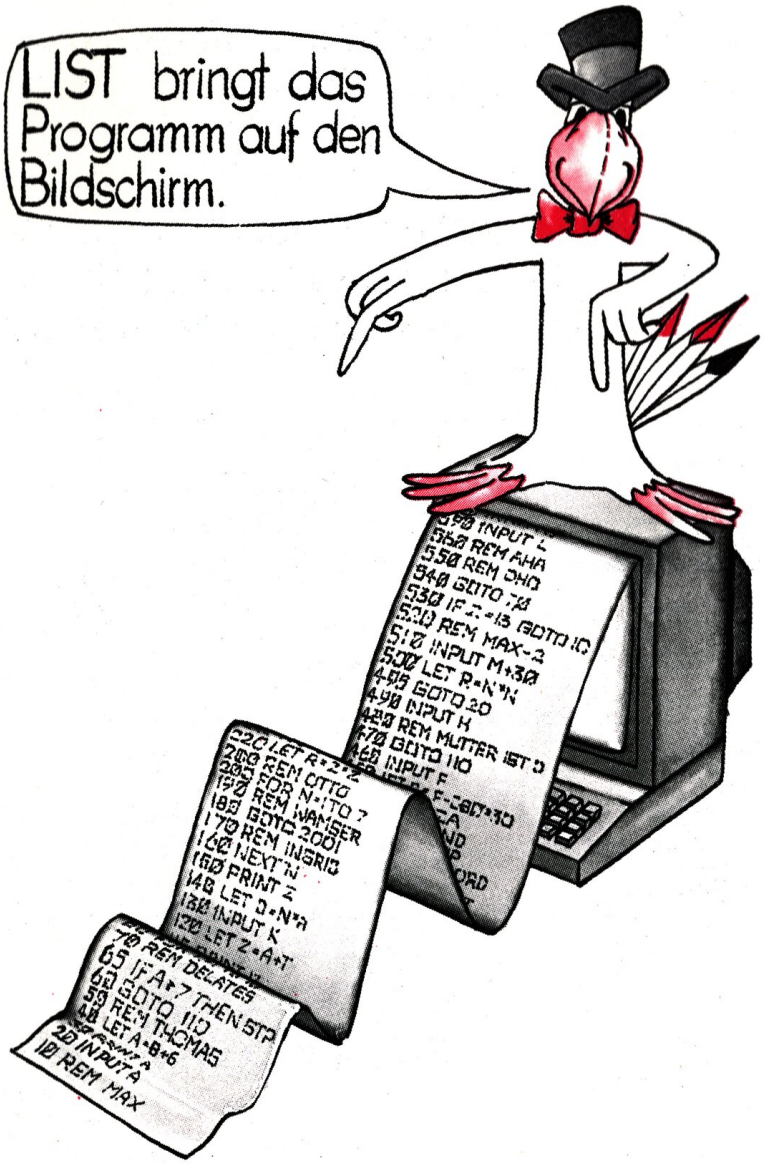
LIST

und drückst die RETURN-Taste. Wenn du alles richtig gemacht hast, müßte nun wieder unser Mini-Programm auf dem Bildschirm stehen. Wenn nicht, versuch es noch einmal.

Auf diese Weise kannst du dein Programm immer wieder aus dem Gedächtnis des Computers auf den Bildschirm holen. Du kannst es dann noch einmal überprüfen und natürlich auch ändern.

Das LIST-Kommando ist besonders wertvoll, falls die Programme länger werden, und wenn das Programm sowie das, was der Computer damit macht, auf dem Bildschirm keinen Platz mehr hat.

Das englische Wort „list" unterscheidet sich kaum



von dem deutschen Wort „Liste“. Man sagt, ein Programm wird „aufgelistet“. Wer alle Zeilen eines Programmes zusammenschreibt, stellt damit ein „listing“ auf.

10 PRINT "COMPUTER“

ist ein „listing“, wenn auch ein kurzes.

Aber eigentlich wollten wir ja einen Weg finden, das Programm zu löschen, ohne dazu den Computer

auszuschalten. Auch hierfür muß man ein besonderes Kommando eintippen. Es heißt

NEW

„New“ (sprich „njuh“) heißt „neu“. Wir sagen damit dem Computer, daß wir etwas Neues anfangen wollen. Und dazu muß der Computer erst mal sein Gedächtnis entrümpeln.

Wenn du mit „NEW“ dem Computer mitgeteilt hast, daß du etwas Neues anfangen willst (natürlich wieder mit der RETURN-Taste), vergißt er schlagartig dein vorheriges Programm.

Das kannst du einfach überprüfen. Weißt du schon wie?

Ja, du läßt dir vom Computer *auflisten,* was er noch im Gedächtnis hat. Dazu gibst du das Kommando

LIST

Nach dem Drücken der RETURN-Taste erscheint auf dem Bildschirm, was noch im Gedächtnis deines Computers steht – nämlich nichts. Er meldet, daß er wieder READY, also bereit ist. Vom Bildschirm ist nach dem NEW-Kommando das Programm *nicht* verschwunden (also genau umgekehrt wie beim „Radieren" der Leertaste.

Um auch noch den Bildschirm „sauber" zu bekommen, verwenden wir unseren alten „Trick": Wir löschen den Bildschirm und schicken zugleich unseren Cursor „heim". Kennst du noch die Taste, mit der wir das schaffen? (Wer's vergessen hat, sieht einfach in der Tabelle auf Seite 116 nach.)

Merk dir auf jeden Fall: Es besteht ein großer Unterschied zwischen dem Leeren des Computer-Bildschirms und dem Leeren des Computer-Gedächtnisses. Bloß beim Aus- und Wiedereinschalten passiert beides gleichzeitig. Aber das soll ab jetzt nur noch unsere letzte „Notbremse" sein.

Jetzt, da das Gedächtnis des Computers „leer" ist (kann man ja mit LIST noch einmal prüfen), schreiben wir unser Programm noch einmal. Diesmal fügen wir aber noch eine Zeile hinzu:

```
10 PRINT "COMPUTER"
20 PRINT "MAX"
```

Wir starten das Programm wieder mit „RUN" und „RETURN". Was der Computer nun „printet", überrascht uns nicht; er führt beide „Befehle" nacheinander aus, dann meldet er sich wieder mit „READY" (fertig, bereit).

Aber jetzt bringen wir mit einer dritten Zeile die Sache gehörig in Schwung: Wir schreiben den „Befehl"

GOTO

„Go to“ ist englisch und heißt: „Gehe nach“ oder „gehe zu“. Wir befehlen damit dem Computer, in eine bestimmte Programmzeile zu springen und dort mit seiner Arbeit weiterzumachen.

Wir schreiben nun die Programmzeile

```
30 GOTO 10
```

Daß zwischen dem ersten Teil des Programms und unserer neuen Zeile ein paar „RUNs“ und „READYs“ stehen, ist völlig egal. Der Computer interessiert sich nur für die *numerierten* Programmzeilen – sofern sie ganz links am Bildschirmrand beginnen.

Kannst du dir schon vorstellen, was passiert, wenn wir unser neues Programm mit „RUN“ und „RETURN“ starten?

Du siehst, das Programm läuft (und wie es läuft!) und will gar nicht mehr aufhören. Denn die Zeile 30 schickt den Computer immer wieder an den Anfang des Programms zurück und läßt es ihn wiederholen und wiederholen und wiederholen . . .

Das läuft so schnell über den Bildschirm, daß man es kaum lesen kann. Aber das stört uns nicht, denn wir haben ja das Programm selbst geschrieben und wissen noch, was drin steht.

Schalte jetzt nicht gleich den Computer aus! Um aus einem Programm „auszusteigen“, benützen wir nämlich auf der Tastatur ganz links eine besondere Taste. Sie hat die Aufschrift

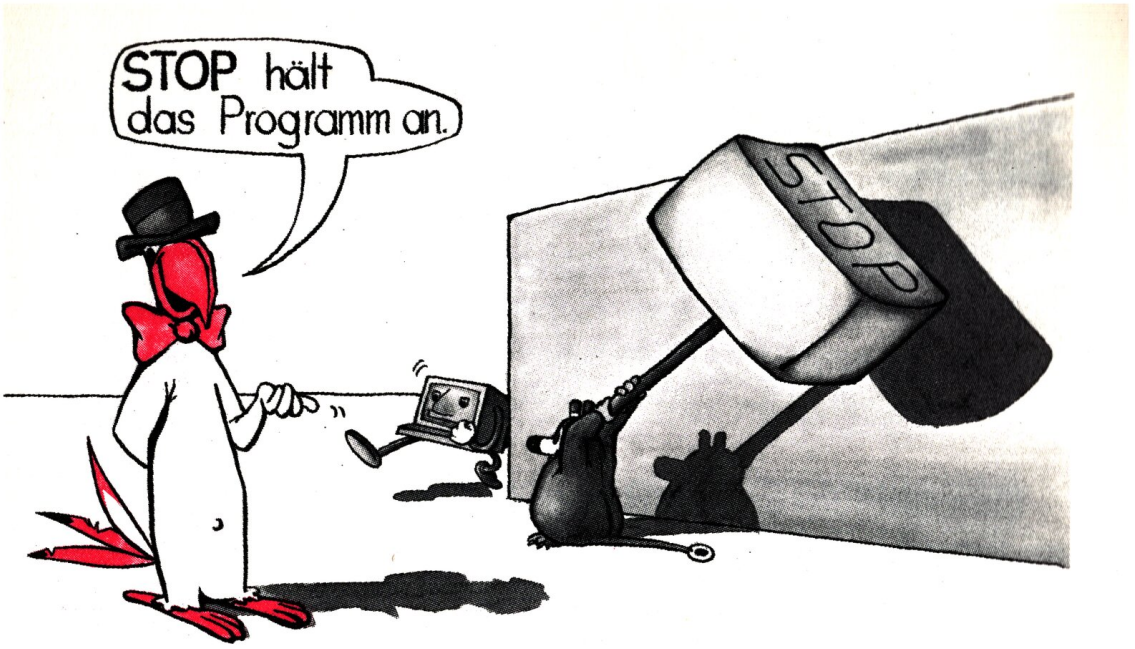
Wenn du „RUN“ eintippst, läuft das Programm los. Betätigst du die Taste RUN/STOP wird es angehalten. Außerdem meldet der Computer, *wo genau* er gestoppt hat:

BREAK IN . . . (Hier steht die Zeilennummer)

Wenn da zum Beispiel

BREAK IN 10

steht, war der Computer gerade mit der Zeile 10 beschäftigt, als er unterbrochen wurde.



Jetzt können wir alle unsere schönen Bildschirmzeichnungen aus Xen oder anderen Zeichen in Programme packen. Wenn wir wollen, können wir sie auch in Bewegung setzen. Dazu geben wir als letzte Programmzeile ein GOTO ein. Dann kehrt der Computer wieder an den Anfang des Programms zurück und legt von vorne los.

Das Aufzeichnen der Bilder auf Kästchen-Papier ändert sich nicht. Denn die Programmzeilen-Nummer und die PRINT-Befehle erscheinen bei *Ausführung* des Programms nicht auf dem Bildschirm. Erst beim Übertragen der Zeichnungen auf den Bildschirm müssen wir umdenken: Wir fangen erst *nach dem Anführungszeichen* zu zählen an. Dabei wird eine Programmzeile manchmal länger als eine Bildschirmzeile. Gut, daß wir eine Kästchen-Schablone gebastelt haben, die uns die Zählerei gehörig erleichtert. Sonst käme man doch ziemlich durcheinander.

Natürlich kann es auch bei diesen „Zeichnungen“ immer wieder vorkommen, daß man „radieren“ muß.

Das geht in einer Programmzeile genauso, wie in einer direkt auf den Bildschirm geschriebenen oder gezeichneten Zeile. Die Verbesserung merkt sich der Computer allerdings erst, wenn du die RETURN-Taste drückst. Mit „RETURN" bringt man ja, wie du weißt, etwas ins Gedächtnis des Computers.

Wenn du mit Hilfe der Cursortasten verbesserst, achte bitte darauf, daß die Anführungszeichen wieder geschlossen sind. Falls auf dem Bildschirm Zeichen erscheinen, die ganz und gar nicht zur Zeichnung passen, tippe erst die letzten Anführungszeichen ein, bevor du mit dem Cursor und der Leertaste ans Radieren gehst. Was passiert, wenn du die Anführungszeichen vergißt, erfährst du im nächsten Kapitel.

Da denkt man: Verbessern mit den Cursortasten ist doch ganz einfach! Man steuert den Cursor auf das falsche X und radiert es mit der Leertaste aus. Dann bringt man den Cursor an die richtige Stelle und schreibt dort das X hin. Probieren wir es aus.

Aber was passiert tatsächlich, wenn du innerhalb eines Programms versuchst, den Cursor zurückzuschicken? Der Cursor geht gar nicht zurück! Stattdessen erscheint ein Zeichen auf dem Bildschirm, das du wahrscheinlich noch nicht gesehen hast. Es ist auch nirgendwo auf der Tastatur zu finden. Das ist so zu erklären: In einem PRINT-Befehl steht zwischen den Anführungszeichen das, was der Computer ausführen oder schreiben soll, *sobald das Programm läuft.* Der Computer schickt also in der PRINT-Zeile nicht den Cursor los, sondern merkt sich, daß er ihn losschicken soll, wenn das Programm läuft.

Im Listing tauchen dann in den Zeilen mit dem

PRINT-Befehl die dazugehörigen „Steuerzeichen“ auf. Das beste ist, man probiert das Ganze mit ein paar „Mini-Programmen“ aus.

Als erstes soll das Programm, sobald es läuft, den Cursor drei Zeilen nach *unten* schicken. Die Programmzeile dafür sieht so aus:

```
10 PRINT "[DOWN][DOWN][DOWN]X"
```

In diesem Programm geht der Cursor drei Zeilen nach unten und legt dort ein "X“ ab.

Wenn der Cursor nach *oben* gehen soll, wird mit der Umschalttaste SHIFT diese Programmzeile geschrieben:

```
10 PRINT "[UP][UP][UP]X"
```

(Cursor nach **oben)**

Nach *rechts* hinaus geht es so:

```
10 PRINT "[RIGHT][RIGHT][RIGHT]X"
```

(Cursor nach **rechts)**

Wie du den Cursor nach *links* steuern kannst, weißt du jetzt bestimmt selber, oder?

```
10 PRINT "[LEFT][LEFT][LEFT]X"
```

(Cursor nach **links)**

Richtig! Genau so!

Aber damit schickst du den Cursor nur in einer Zeile hin und her. Nach oben springt der Cursor leider nicht automatisch am Ende einer Zeile zur nächsthöheren, wie er das nach unten zur jeweils

nächsttieferen tut, sobald er am Ende der Zeile angelangt ist.

Schick deinen Cursor zur Übung ein bißchen auf dem Bildschirm herum. Du kannst dafür auch mehrzeilige Programme schreiben.

Nur ein paar Beispiele:

```
10 PRINT"[CRSR DOWN][CRSR DOWN][CRSR DOWN]XX[CRSR RIGHT][CRSR RIGHT][CRSR RIGHT]XX"
20 GOTO 10
```

In diesem Programm legt der Cursor zunächst zwei X-en 3 Zeilen weiter unten ab und zwei weitere X-en zwei Zwischenräume weiter rechts. Die X-en zeigen uns, welchen Weg der Cursor genommen hat.

Zwischen den einzelnen Programmen löschen wir mit „NEW“ das Gedächtnis des Computers, damit er nicht durcheinanderkommt.

Mit diesem kleinen Programm wird zehn Zeilen weiter unten ein Viereck gezeichnet:

```
10 PRINT "[CRSR DOWN][CRSR DOWN][CRSR DOWN][CRSR DOWN][CRSR DOWN][CRSR DOWN][CRSR DOWN][CRSR DOWN][CRSR DOWN][CRSR DOWN]XXXX"
20 PRINT "X   X"
30 PRINT "X   X"
40 PRINT "XXXX"
```

Dieses Programm ist ein Witz (überleg erst, was dabei herauskommt, und probier es dann aus):

```
10 PRINT "[CRSR RIGHT][CRSR RIGHT][CRSR DOWN][CRSR DOWN][CRSR DOWN][CRSR RIGHT][CRSR RIGHT][CRSR RIGHT][CRSR UP][CRSR UP][CRSR UP]X"
RUN
X

READY.
```

(Dieses Programm schickt den Cursor einmal im Kreis herum.)

Mit diesen Steuerzeichen in einem Programm zu arbeiten macht Spaß. Man kommt aber auch leicht durcheinander, weil man zum Beispiel die Cursor-Steuerung nach oben und unten nur in einer Zeile eingeben muß, die nach rechts und links aber für jede Zeile neu. Wenn wir zum Beispiel ein kleines Viereck nach rechts unten versetzen wollen, sieht das Programm so aus:

```
READY.
10 PRINT "[CRSR DOWN][CRSR DOWN][CRSR RIGHT][CRSR RIGHT]XXXX"
20 PRINT "[CRSR RIGHT][CRSR RIGHT]X[CRSR RIGHT][CRSR RIGHT]X"
30 PRINT "[CRSR RIGHT][CRSR RIGHT]X[CRSR RIGHT][CRSR RIGHT]X"
40 PRINT "[CRSR RIGHT][CRSR RIGHT]XXXX"
RUN


    XXXX
    X  X
    X  X
    XXXX

READY.
```

In der ersten Zeile haben wir die Bewegung nach unten festgelegt, und in jeder Zeile mußte die Bewegung nach rechts neu geschrieben werden. (Das ist verständlich, wenn man bedenkt, daß der Computer nach jedem „RETURN“ an den Anfang der nächsten Zeile geht.)

Versuch doch einmal, ein solches Viereck ganz nach unten und möglichst weit nach rechts zu bringen. Du weißt ja bestimmt noch, wieviel Zeilen dein Computer hat, und wieviel Zeichen in eine Zeile passen. Deine Kästchen-Schablone hilft dir dabei.

```
ZUR ERINNERUNG:

"■"= DIE CLR/HOME-TASTE MIT SHIFT.
     MACHT (IM PROGRAMM) DEN BILDSCHIRM
     SAUBER.
"■"= CURSOR NACH UNTEN
"■"= CURSOR NACH OBEN
"■"= CURSOR NACH RECHTS
"■"= CURSOR NACH LINKS
```

Wenn wir in unseren Bildschirm-Zeichnungen etwas zu löschen haben, benützen wir die Leertaste. Oft ist es aber zweckmäßiger, eine andere Radier-Methode anzuwenden.

Wir beschreiben diese andere „Radier-Methode" erst jetzt, weil sie nicht die einfachste ist, und man ein bißchen Gefühl dafür braucht, wie der Computer auf den Bildschirm schreibt. Aber jetzt hast du ja den Bildschirm „erobert" und bewegst dich darauf bestimmt schon sicherer als so mancher alte Programmier-Hase.

Lösche bitte wieder das Gedächtnis des Computers und den Bildschirm. Und dann schreibe diesen Satz:

DAZ IST EINN FALZER SATSCH

So ein Unsinn! Das radieren wir aber sofort wieder aus! Ganz oben rechts ist die Taste mit der Aufschrift

INST
DEL

Das sind natürlich wieder Abkürzungen von englischen Wörtern. Für uns reicht es im Augenblick,

daß diese Taste unser „schneller Löscher“ ist. Was passiert, wenn du die Taste einmal drückst?

Weg ist der Buchstabe. Und was passiert, wenn du die Taste heruntergedrückt läßt?

Der „FALZE SATSCH“ ist schneller weg, als wir ihn geschrieben haben. Welche Eigenschaft hat die INST/DEL-Taste? Auch die Leertaste hat diese Eigenschaft:

INST/DEL hat Dauerfunktion

Mit der INST/DEL-Taste, unserem „schnellen Löscher“, können wir in einer Zeile also von rechts nach links radieren. (Mit der Leertaste löschen wir von links nach rechts.)

Das ist sehr praktisch. Wenn man gerade schreibt und merkt, daß das letzte Wort oder die letzten Zeichen falsch sind, radiert man sie schnell von hinten weg, und muß nicht erst mit dem Cursor auf Wanderschaft gehen.

Man kann auch mitten in einem Satz mit der INST/DEL-Taste radieren.

Schreiben wir noch einmal einen „Falzen Satsch“.

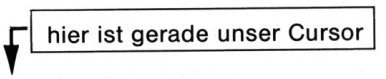
Zum Beispiel:

COMPUTER-MAZ IST NICHT DA

Um aus dem MAZ einen MAX zu machen, müssen wir mit dem Cursor an die richtige Stelle. Aber was ist die richtige Stelle?

Mit unserem „schnellen Löscher" radieren wir von rechts nach links. Also müssen wir den Cursor rechts neben den Buchstaben stellen, den wir löschen wollen.

In dem Satz



COMPUTER-MAZ■ IST NICHT DA

steuern wir den Cursor genau zwischen MAZ und IST. Und jetzt radieren wir das Z durch einen Druck auf die INST/DEL-Taste aus.

Aber dabei passiert noch etwas anderes: Unser Löscher radiert den falsch geschriebenen Buchstaben, holt aber auch die nachfolgenden Buchstaben um einen Platz zu sich her. (Und wenn man nicht aufpaßt, frißt er wie ein wildgewordener kleiner Pac-Man viel mehr von unserem Text, als uns lieb ist!)

Jetzt haben wir keinen Platz mehr, um den richtigen Buchstaben einzusetzen. Wir müssen uns diesen Platz erst schaffen.

Das machen wir auch mit der INST/DEL-Taste, die wir zusammen mit der Umschalttaste SHIFT drükken.

Falls dein Cursor gerade an der richtigen Stelle ist, probier es einfach aus. Sonst kannst du ihn mit der Cursorsteuerung dorthin bringen.

„INST/DEL" und „SHIFT" rückt mit jedem Tastendruck den Text rechts vom Cursor um eine Stelle nach rechts und schafft Platz für Berichtigungen.

Wenn wir das in einer PRINT-Zeile in einem Programm machen, müssen wir darauf achten, daß die Anführungszeichen stehenbleiben. Sonst funktioniert's nicht!

Also gehen wir zum Beispiel bei

```
10 PRINT "COMPUTER-MAZ IST
NICHT DA"
```

nach dem Anführungszeichen auf den Fehler mit der Cursorsteuerung zu, *nachdem* wir die Anführungszeichen geschlossen und den PRINT-Befehl mit „RETURN“ gesichert/abgespeichert haben.

Da kann man schon ganz schön durcheinanderkommen. Am Anfang brauchst du bestimmt eine Menge Versuche, bis du die richtige Art der Verbesserung gefunden hast.

Am besten ist es vorerst, du verwendest die Berichtigungsmethode, die dir am leichtesten fällt – auch wenn sie nicht unbedingt die eleganteste ist.

Im letzten Beispielsatz hätten wir es uns mit der Berichtigung sowieso viel einfacher machen können. Ist dir das auch schon aufgefallen?

Der Computer schreibt ja ein neues Zeichen immer über das alte. Also hättest du den Cursor nur auf das Z steuern und ein X darüber schreiben müssen. (In einer Programmzeile muß noch „RETURN“ gedrückt werden, damit der Computer die Berichtigung auch in sein Gedächtnis nimmt.)

Dafür wissen wir jetzt, was man mit dem „schnellen Löscher" alles anfangen kann. Wir haben mit der INST/DEL-Taste überflüssige Buchstaben gelöscht, Text herangezogen – und wieder weggedrückt, um Platz zu schaffen. (Für das „Wegdrücken" haben wir noch die SHIFT-Taste gebraucht.)

Nun kennen wir drei Möglichkeiten zu berichtigen:

- Über etwas Falsches oder Überflüssiges kannst du das Richtige darüberschreiben. (Achte darauf, daß rechts keine Zeichen vom alten Text übrigbleiben.)
- Mit der Leertaste kannst du von links nach rechts radieren, nachdem du den Cursor an die richtige Stelle gesteuert hast. (Dabei entstehen Lücken, die du mit der INST/DEL-Taste wieder schließt.)
- Mit der INST/DEL-Taste kannst du von rechts nach links radieren. (Dabei wird der nachfolgende Text herangezogen. Mit INST/DEL und der SHIFT-Taste schiebst du ihn wieder weg.)

Uff, das ist ganz schön vertrackt. Da kommen auch alte Computer-Hasen leicht ins Schleudern. Aber wir haben jetzt so viele Verbesserungsmethoden gelernt, daß wir jeden Fehler auf die eine oder andere Art beseitigen können.

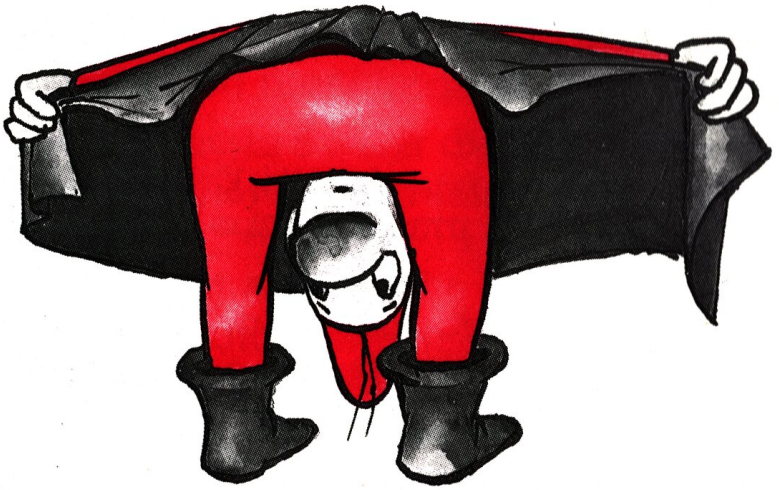
Am besten probierst du damit so viel wie möglich herum. Und das macht natürlich am meisten Spaß, wenn du ein Spiel daraus machst:

Einer schreibt zum Beispiel einen Satz mit einem oder mehreren Fehlern, und der andere muß die Fehler finden und berichtigen.

So ein Spiel kannst du auch gut mit dir selbst

spielen. Du tippst ein kurzes Wort mit geschlossenen Augen ein. Vorher prägst du dir ein, wo die Tasten dafür sind. Und wenn in dem Wort Fehler stecken, verbesserst du sie.

Wie berichtigt man am elegantesten einen Satz wie



MEIN COMPUTER UNT ICH

– den Cursor auf das „T" steuern und das „D" über das T tippen

und wie

MEIN COMPUTER UNDT ICH

– den Cursor vor das „I" vom „ICH" steuern und dann den „schnellen Löscher" (INST/DEL) betätigen.

oder

MEINCOMPUTER UND ICH

– den Cursor auf das „C" von „COMPUTER" steuern und die INST/DEL-Taste zusammen mit „SHIFT" drücken.

(Die Antworten sind wieder auf den Kopf gestellt, aber wahrscheinlich brauchst du sie gar nicht mehr.)

Hier ist noch etwas Kniffliges: Wohin mußt du den Cursor bei dem Fehler-Satz

MEIN COMPUTER UNND ICH

steuern, wenn du das UNND mit der INST/DEL-Taste verbessern willst? (Dabei ist es egal, ob du das erste oder das zweite „N“ ausradierst.) Das bekommst du am besten durch Ausprobieren heraus.

Der Cursor muß entweder auf das zweite „N“ oder auf das “D“ vom “UNND“.

Besonders zu zweit oder zu dritt macht es Spaß mit dem folgenden Spielchen das Radieren auf dem Bildschirm zu üben.

In unserem **STERNCHEN-SPIEL** schreibt einer dem anderen einen Satz, in den er ein paar Sternchen einbaut. Die Taste mit den Sternchen hast du sicher schon gefunden. (Beim Commodore ist es die dritte von rechts in der zweiten Zeile.)

```
FIN*DIE DI*E STERNCHEN*,
MAEX*CHEN!
```

Jetzt soll der zweite Spieler die Sternchen herausholen und möglicherweise entstehende Lücken wieder schließen.

Aha, da braucht er wohl die INST/DEL-Taste!

Er hat eine Minute Zeit (ein Profi nur eine halbe).

Für jedes nicht gelöschte Sternchen bekommt er einen Minuspunkt. Und wenn er einen Buchstaben aus Versehen gelöscht und nicht wieder hinein bekommen hat, bekommt er zwei Minuspunkte. Das ist manchmal ganz schön vertrackt. Wie kriegt man zum Beispiel dieses Sternchen-Wort richtig hin:

STER***EN

Hier sind noch einige Aufgaben-Beispiele, an denen man zu mehreren oder allein seine Geschicklichkeit beim Verbessern testen kann (links stehen die Fehler und rechts sind sie verbessert):

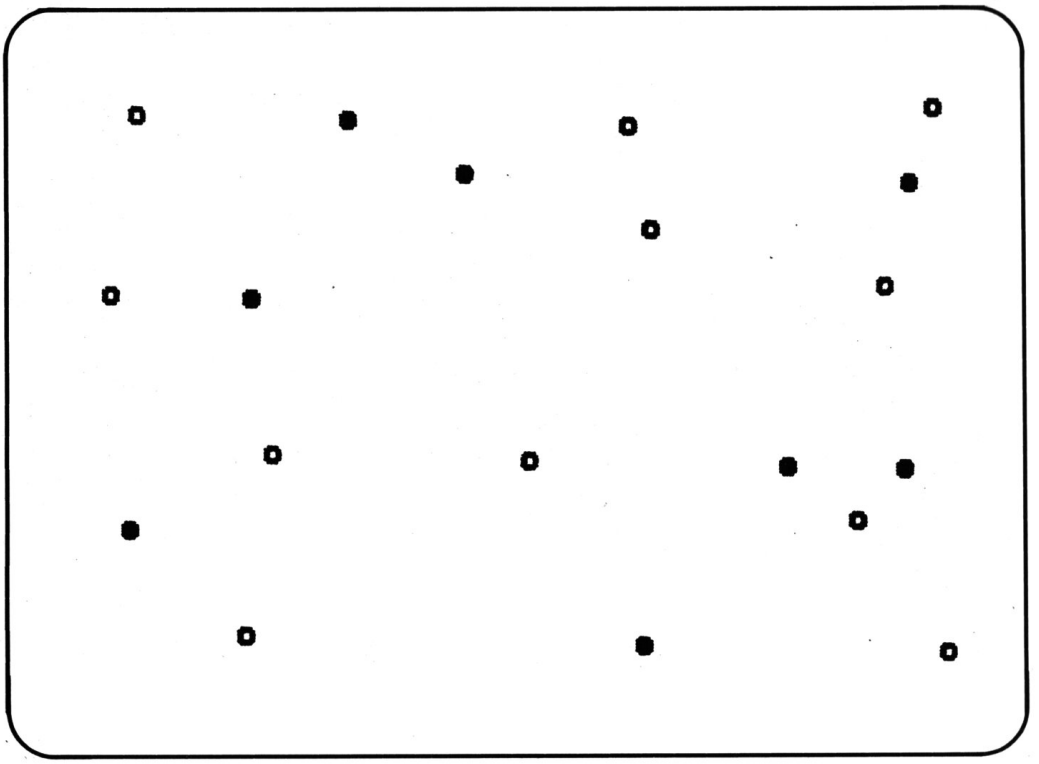
KAE SEBROT	KAESEBROT
MAX WIRD LANGSAMM MUEDE	MAX WIRD LANGSAM MUEDE
3+4=6	3+4=7
STRASSENBAN	STRASSENBAHN

Weil wir jetzt beim Verbessern so gut zu Rande kommen, können wir ein paar kniffligere Spiele mit der Cursor-Steuerung veranstalten.

Verteile einmal über den ganzen Bildschirm volle und leere Kreise. Die schreibt man mit den Tasten Q und W und „SHIFT".

Setze die Kreise aber nicht zu nah aneinander!

Dein Bildschirm könnte jetzt so aussehen:



Nun bringe deinen Cursor nach Hause.

Das Spiel besteht darin, das Bild auf dem Schirm möglichst schnell zu verändern: An die vollen Kreise soll oben und unten je ein Herzchen gehängt werden (die Herzchen schreibt man mit der S- und der SHIFT-Taste); an die leeren Kreise soll links und rechts je ein Karo mit der „Z"- und SHIFT-Taste gehängt werden.

Du machst also aus dem

eine solche Figur:

Aus dem

wird eine solche Figur

Um an die richtigen Stellen zu gelangen, verwenden wir wieder die Cursor-Steuerung. Falls etwas danebengegangen ist, verbessern wir es möglichst schnell.

Also: Auf die Plätze (der Cursor wird heimgeschickt)

Fertig (die Figur auf „SHIFT“ und die Cursor-Steuerung)

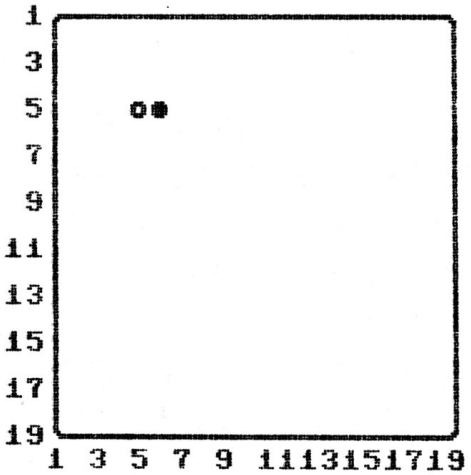
Los!

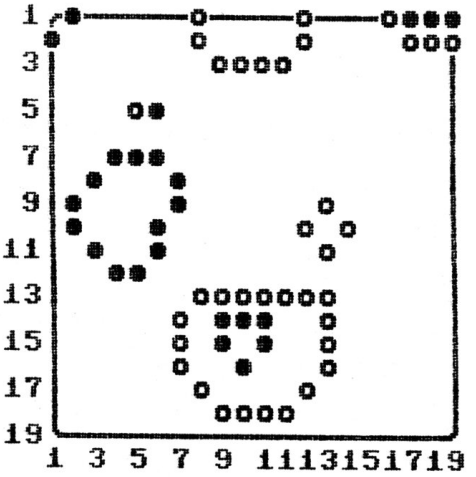
Wer schafft es am schnellsten?

Auf dem Bildschirm können wir auch eine Variante von „GO“ spielen. GO ist uralt und stammt aus China.

Dabei „zeichnet“ ein Spieler weiße (leere) Kreise auf den Bildschirm, der andere schwarze (volle). Jeder Spieler versucht nun, den Gegner mit seinen eigenen Kreisen einzuschließen. Die Kreise sollen also nicht **auf,** sondern **um** die des Gegners **herum** gelegt werden.

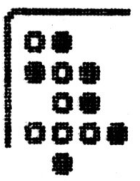
Wir fangen irgendwo in der Bildschirmmitte an. Es dürfen nicht mehr als acht Steine in einer Reihe liegen. Wenn ein „Stein“ von denen des Gegners eingeschlossen wurde, wird er gelöscht. Wer dabei allerdings einen Fehler macht, hat verloren. Gewinner ist, wer die meisten gegnerischen „Steine“ eingeschlossen und richtig gelöscht hat.

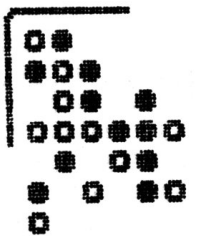




(GO)

Ein ähnliches Spiel macht mit dem Computer genausoviel Spaß wie auf dem Papier: GOBANG.





Sieg für Schwarz.

Und nun machen wir einen kleinen Computerfilm. Zuerst zeichnen wir mit unserer Kästchen-Schablone einen Pfeil aus Xen. Diesen Pfeil verpacken wir in ein Programm. Das sieht dann so aus:

```
10 PRINT"   X  "
20 PRINT"  XX  "
30 PRINT" XXX  "
40 PRINT"   X  "
50 PRINT"   X  "
60 PRINT"   X  "
70 PRINT"   X  "
80 PRINT"   X  "
90 PRINT"   X  "
```

Sobald du das Programm mit „RUN“ und der RETURN-Taste startest, erscheint ein hübscher Pfeil auf dem Bildschirm. Das ist eigentlich nichts Besonderes; wir haben ja schon ganz andere Sachen auf den Bildschirm gebracht.

Aber wir wenden jetzt noch einen besonderen Trick an. Der Pfeil soll nämlich immer allein auf dem Bildschirm stehen, egal, wie oft wir das Programm laufen lassen: Nie sollen Reste von alten Texten auf dem Bildschirm im Weg herumstehen.

Das Werkzeug, mit dem wir den Bildschirm „säubern“ und den Cursor nach Hause links oben schikken, ist die CLR/HOME-Taste.

Wie bekommen wir eine neue Zeile **vor** unser bisheriges Programm?

Nun zeigt es sich, warum es wichtig ist, daß wir unser Programm in Zehnerschritten numerieren. Denn wir müssen jetzt nur eine Zeilennummer eintippen, die niedriger als 10 ist. Der Computer stellt die dazugehörige Zeile dann automatisch an den Anfang des Programms.

Dabei ist es egal, wann du diese Zeile schreibst. Das kann geschehen (nach vielen Versuchen mit dem Pfeil zum Beispiel), wenn das Programm gar

nicht mehr zu sehen ist. Genausogut kann man diese Zeile direkt unter dem Programm eingeben. Der Computer arbeitet die Befehls-Nummern ihrer Größe nach ab, gleichgültig, in welcher Reihenfolge sie eingegeben worden sind. Achte nur darauf, daß die neue Programmzeile ganz links beginnt.

Und jetzt lassen wir unseren Pfeil losfliegen.

Dazu fügen wir unserem Programm noch ein paar Zeilen hinzu. In ihnen passiert eigentlich gar nichts, sie bleiben leer. Wenn aber diese leeren Befehls-Zeilen den unteren Bildschirmrand erreichen, schieben sie den Pfeil nach oben aus dem Bild. Das haben

wir schon gesehen, als wir bei vollem Bildschirm weitergeschrieben haben.

Der Befehl dafür, eine Zeile frei zu lassen, ist ein „PRINT“ ohne etwas dahinter (also auch ohne Anführungszeichen).

Schreiben wir einmal eine Reihe von solchen Programmzeilen. Also

```
100 PRINT
110 PRINT
120 PRINT
130 PRINT
140 PRINT
150 PRINT
160 PRINT
170 PRINT
180 PRINT
190 PRINT
200 PRINT
210 PRINT
220 PRINT
230 PRINT
240 PRINT
250 PRINT
```

Und nun starten wir das Programm.

Der Pfeil fliegt davon!

Und er bleibt weg.

Wie können wir ihn zurückholen?

Der Pfeil erscheint wieder, wenn wir mit „RUN“ und der RETURN-Taste das Programm noch mal starten.

Und wieder ist er weg.

Statt unser Programm nun jedesmal neu zu starten, können wir dem Computer innerhalb des Programms befehlen, dieses Programm selbsttätig zu

wiederholen, sobald er es abgearbeitet hat.

Der Befehl dazu ist „GOTO“ und dann die Nummer der Programmzeile, mit der es weitergehen soll. Wir fügen also unserem Programm noch eine Zeile hinzu:

```
260 GOTO 10
```

Der Computer weiß jetzt, daß er jedesmal, wenn er am Ende des Programms angelangt ist, zurück zur Zeile 10 gehen und dort wieder mit dem Programm anfangen soll.

Wenn wir jetzt das Programm starten, haben wir eine „unendliche Geschichte“ als Computerfilm! Der Computer würde so in aller Ewigkeit weitermachen, wenn wir ihm nicht einmal mit der RUN/STOP-Taste sagen würden, daß er aufhören soll.

In einem solchen Computerfilm können wir zum Beispiel einen Hubschrauber fliegen lassen. So sieht das Programm dazu aus:

```
1040 REM*HUBSCHRAUBER*
1050 PRINT"            =====@=====
1060 PRINT"                 I
1070 PRINT"                 I
1080 PRINT"              [▒▒▒▒▒]
1090 PRINT"             ▒ ▒ ▒
1100 PRINT"             ▒ ▒ ▒▒▒▒▒▒▒*
1110 PRINT"             [▒▒▒▒]▒▒▒▒▒
1120 PRINT"               OO      O
1130 PRINT
1140 PRINT
1150 PRINT
1160 PRINT
1170 PRINT
1180 PRINT
```

```
1190 PRINT
1200 PRINT
1210 PRINT
1220 PRINT
1240 GOTO1050

READY.
```

Verschiedene Sterne im Weltall, ein Raumschiff, einen Schwarm Vögel, alles, was dir einfällt, kannst du davonfliegen lassen.

Eine ähnliche Fehlersuche wie beim **STERNCHEN-SPIEL** findet in dem Spiel statt, das jetzt als Belohnung und Übung kommt.

Das **RECHTSCHREIB-SPIEL.**

Du kannst dieses Spiel entweder von der Diskette laden oder von Freunden aus dem Listing im Anhang dieses Buches eintippen lassen. Den Namen dieses Spiels findest du im Inhaltsverzeichnis der Diskette.

Das Inhaltsverzeichnis heißt „S“ und wird mit dem Kommando LOAD "S", 8 von der Diskette in den Computer geladen.

Der Computer zeigt dir einen Text, in dem es ein paar Lücken gibt. Er zeigt immer mit einem Pfeil auf die Lücke in einem Wort, die du füllen sollst. Du drückst dazu einfach auf die Taste mit dem Buchstaben, den du jeweils für den richtigen hältst. Der Computer sagt dir zum Schluß, welche Worte du richtig ergänzt hast, und bittet dich, es bei den anderen noch einmal zu versuchen.

```
   >  > >> RECHTSCHREIBSPIEL << <  <

IN DIESEM SPIEL GEHT ES DARUM, EINEN
DURCH DEN COMPUTER ENTSTANDENEN LUECKEN-
TEXT AUSZUFUELLEN. MAN KANN DABEI IMMER
NUR DIE LUECKE BEARBEITEN, UNTER DER DER
PFEIL STEHT. DEN PFEIL KANN MAN MIT DEN
CURSORSTEUERTASTEN BEWEGEN. ES SIND NUR
BUCHSTABEN ZU ERGAENZEN.

FERTIG? DRUECKE BITTE IRGENDEINE TASTE!
```

Quizmaster, Codeknacker und Gedächtniskünstler

Unser Computer ist auch ein Rechner. Das wollen wir jetzt ausnützen. Aber keine Angst: Es soll vor allem Spaß machen.

Du kannst deinen Computer verwenden wie einen Taschenrechner. Das heißt, du stellst eine Aufgabe, und die Maschine löst sie.

Ein Computer kann die komplizierteste Aufgabe schneller lösen als der gescheiteste Rechen-Professor. Und wenn du die Aufgabe richtig stellst, kannst du ganz sicher sein, daß der Computer eine fehlerfreie Antwort gibt.

Wenn der Computer die richtige Lösung nicht findet, liegt es immer an der Frage oder am Programm. Verrechnen wie ein Mensch kann er sich nicht.

Wir haben den Computer schon etwas auf den Bildschirm schreiben lassen mit dem Befehl

PRINT

Derselbe Befehl sagt dem Computer aber auch, daß er eine Rechenaufgabe lösen soll. Nur darf man nun keine Anführungszeichen mehr eingeben.

PRINT 3 + 4

und betätige dann die RETURN-Taste.

Sofort zeigt der Computer das Ergebnis an (nämlich 7) und meldet sich wieder „READY".

Was passiert im Unterschied dazu, wenn du die Anführungszeichen doch setzt (wie bei unserem

Schreiben und Zeichnen)? Probieren wir es aus: Tippe einmal

PRINT „3 + 4“

Du siehst: Wenn du den PRINT-Befehl unter Anführungszeichen eingibst, druckt der Computer genau das, was zwischen diesen Anführungszeichen steht.

Sobald du dagegen den PRINT-Befehl ohne Anführungszeichen gibst, versucht der Computer, die Aufgabe nach dem „PRINT“ zu lösen.

Daraus machen wir jetzt ein kleines Programm, bei dem es vor allem darauf ankommt, wo wir Anführungszeichen setzen und wo nicht.

```
10 PRINT "DIESE RECHNUNG"
20 PRINT "IST LEICHT:"
30 PRINT "3+4=?"
40 PRINT "NA SAG SCHON!"
50 PRINT "3+4="; 3+4
```

Und das siehst du dann auf dem Bildschirm, wenn der Computer das Programm ausführt:

```
10 PRINT "DIESE RECHNUNG"
20 PRINT "IST LEICHT:"
30 PRINT "3+4=?"
40 PRINT "NA SAG SCHON!"
50 PRINT "3+4="; 3+4
RUN
DIESE RECHNUNG
IST LEICHT:
3+4=?
NA SAG SCHON!
3+4= 7

READY.
```

Wir müssen noch das Zeichen [;] (Strichpunkt oder Semikolon) erklären. Es sagt dem Computer,

daß er den zweiten Teil des PRINT-Befehls gleich hinter den ersten drucken soll.

Anders wäre es, wenn wir da ein Komma eintippen würden. Dann sähe es auf dem Bildschirm so aus:

```
10 PRINT "DIESE RECHNUNG"
20 PRINT "IST LEICHT:"
30 PRINT "3+4=?"
40 PRINT "NA SAG SCHON!"
50 PRINT "3+4=", 3+4
RUN
DIESE RECHNUNG
IST LEICHT:
3+4=?
NA SAG SCHON!
3+4=          7

READY.
```

Und so sähe es aus, wenn gar kein Zeichen zwischen „3+4“ und 3+4 stünde:

```
10 PRINT "DIESE RECHNUNG"
20 PRINT "IST LEICHT:"
30 PRINT "3+4=?"
40 PRINT "NA SAG SCHON!"
50 PRINT "3+4="3+4
RUN
DIESE RECHNUNG
IST LEICHT:
3+4=?
NA SAG SCHON!
3+4= 7

READY.
```

In diesem Programm kannst du nun immer wieder andere Zahlen, aber auch andere Rechenarten einsetzen. Du bringst dazu das Programm mit

LIST

auf den Bildschirm und änderst die Zahlen beziehungsweise die Rechenart entweder durch darüberschreiben oder durch die anderen dir bekannten

Berichtigungsarten. (Vergiß nicht, die Änderung mit „RETURN“ ins Gedächtnis deines Computers zu bringen.)
Die Rechenzeichen auf deinem Computer sind:

⊞ Das Und- oder Pluszeichenzeichen kennst du aus der Schule.

⊟ Das Weniger- oder Minus-Zeichen findest du auf dem Computer gleich neben „+“. Es dient zugleich auch als Bindestrich und als Abkürzung für das Wort „bis“ („1–2“ heißt dann „eins bis zwei“).

⊡ Das Sternchen, das wir schon in unserem Fehlersuchspiel verwendet haben, *ist auf dem Computer auch das Mal-Zeichen.* **Wir** schreiben für „mal“ sonst meistens einen Punkt, manchmal auch ein „x“.
Dein Computer kennt als Mal-Zeichen nur das „*“

⊠ Der schräge Strich ist das Zeichen für „geteilt durch“. Du findest dieses Zeichen auf derselben Taste, auf der auch das „?“ steht. (Um das Fragezeichen zu schreiben, mußt du gleichzeitig auf die SHIFT-Taste drücken.)

Auf dem Papier benutzen wir beim Rechnen für „geteilt durch“ den Doppelpunkt oder den Bruchstrich; ($12:3 = \frac{12}{3}$; beides heißt „zwölf geteilt durch drei“)

Der Computer versteht aber nur den schrägen Strich (/) als Geteilt-durch-Zeichen, also 12/3.

Das kannst du jetzt ein bißchen üben, indem du den Computer mit PRINT ein paar Rechnungen lösen läßt.

Dabei verwenden wir allerdings keine Zeilen-Nummern. Wir sprechen mit dem Computer in „Direkt-Modus“. So nennt man es, wenn der Computer eine Aufgabe direkt löst, ohne auf ein ganzes Programm und das Kommando „RUN“ zu warten.

```
PRINT 12 * 2
```

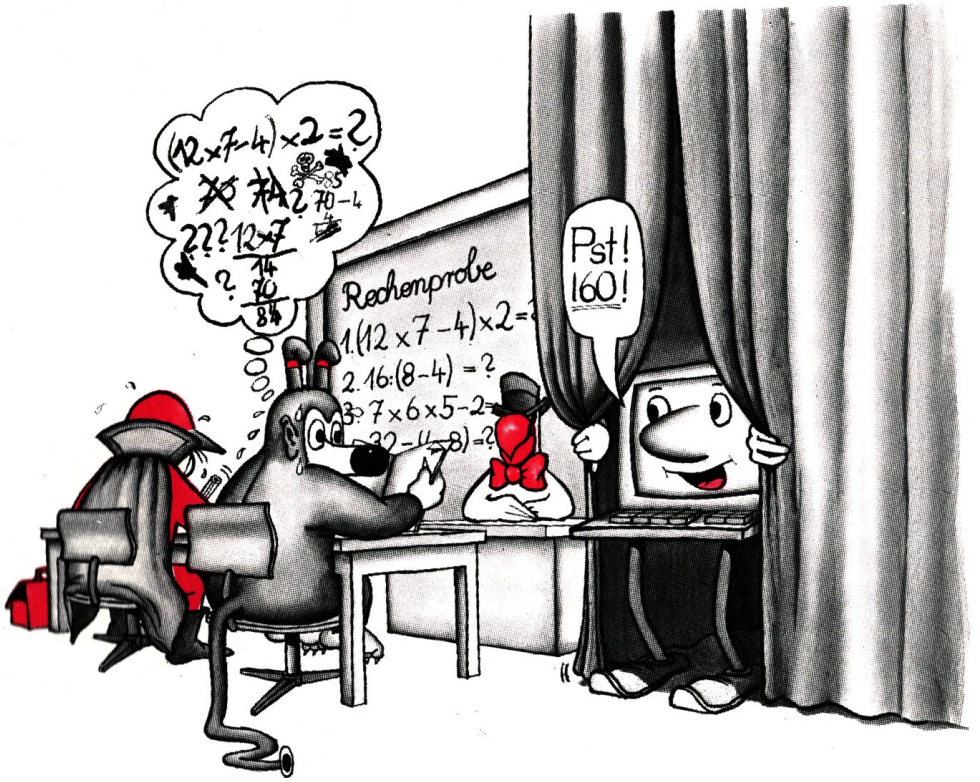
kann dir der Computer ebenso ausrechnen wie zum Beispiel

```
PRINT 10 / 2
```

Wer es schon gelernt hat, kann es auch mit der Kettenrechnung versuchen:

```
PRINT 2 * (3+4)
```

Natürlich kannst du ab jetzt alle Rechenaufgaben von deinem Computer lösen lassen. Das wäre aber ziemlich langweilig und auch unfair.



Viel spannender ist es, wenn du erst einmal versuchst, deine Rechnungen selbst zu lösen und dann den Computer als Quizmaster einsetzt. Er kann dir nämlich auch sagen, ob deine Lösung richtig oder falsch ist.

Das geht ganz einfach. Du gibst dem Computer deine Lösung ein, und er antwortet mit der Meldung falsch oder richtig.

Wenn dein Ergebnis falsch ist, zeigt dir das der Computer mit einer „0“ an.

Wenn du die Rechnung richtig gelöst hast, antwortet der Computer mit „–1“.

Du fragst zum Beispiel

```
PRINT 3+4=8
```

Darauf antwortet der Computer

```
 0
```

Klar, drei plus vier ist sicher nicht acht.
Fragst du dagegen

```
PRINT 3+4=7
```

antwortet der Computer

```
-1
```

So sieht es auf dem Bildschirm aus:

```
PRINT 3+4=8
 0

READY.
PRINT3+4=7
-1

READY.
```

Ich kann mir vorstellen, daß dir auf diese Art die Rechenaufgaben mehr Spaß machen als bisher.

Aber bitte laß nicht alles den Computer machen. *Du* willst ja rechnen lernen. Das gelingt dir, wenn du *mit* dem Computer arbeitest, nicht, wenn du ihn *für dich* arbeiten läßt.

Jetzt wollen wir einmal das Gedächtnis unseres Computers prüfen. Nebenbei prüfen wir auch unser eigenes.

Wir geben zum Beispiel diese Behauptung ein:

```
A=3
```

Damit haben wir eigentlich nur der Zahl drei einen anderen Namen gegeben. Wenn dir „MAX" als neuer Name für die Drei besser gefällt, schreib

```
MAX=3
```

Nachdem wir mit „RETURN" unsere Umbenennung im Gedächtnis des Computers verankert haben, steht auf dem Bildschirm:

```
MAX=3

READY.

PRINT MAX
 3

READY.
```

Jetzt löschen wir den Bildschirm. Aber tippe dazu auf keinen Fall „New" ein und schalte nicht einfach den Computer aus, denn damit würdest du nicht (nur) den Bildschirm, sondern das Gedächtnis des Computers löschen! Aber das weißt du ja längst . . .

Und jetzt schauen wir nach, ob sich der Computer noch an die Zahl erinnert, die wir dem Wort „MAX" zugeordnet haben. Wir fragen ihn

PRINT MAX

und betätigen anschließend natürlich die RETURN-Taste. Der Computer beginnt zu arbeiten und in seinem Gedächtnis zu kramen.

Er antwortet: 3

Wir könnten statt

MAX=3

genauso schreiben

MAX=33
MAX=56

oder

MAX=36701897

Du kannst also jede beliebige Zahl in „MAX" umtaufen. Der Computer benutzt in seinem Gedächtnis von dem neuen Namen allerdings nur die ersten zwei Buchstaben. Ihm ist es also egal, ob wir „MAX" oder „MATTERHORN" oder einfach nur „MA" eingeben. Man nennt einen solchen Namen für eine Zahl, der entweder aus einem oder aus zwei Buchstaben besteht, eine *Variable.*

Für unser Spiel wollen wir längere Namen verwenden. Den Computer stört das nicht.

Nun geben wir einer zweiten Zahl einen anderen Namen, zum Beispiel

COMPUTER=4

Dem Wort MAX haben wir anfangs die Drei zugeordnet:

MAX=3

So sieht es dann auf dem Bildschirm aus:

```
MAX =3

READY.
COMPUTER =4

READY.
PRINT MAX+COMPUTER
 7

READY.
```

Wir löschen den Bildschirm wieder, und nun lassen wir den Computer die beiden Zahlen zusammenzählen, die wir in „MAX“ und „COMPUTER“ umbenannt haben, also addieren:

PRINT MAX+COMPUTER

Der Computer antwortet prompt: 7

Daraus kann man auch ein Spiel für zwei machen. Der erste Spieler benennt zwei Zahlen um (nachdem er „NEW“ eingegeben hat), zum Beispiel

COMPUTER=2

und

MAX=8

Jetzt löscht dieser Spieler den Bildschirm und läßt den Computer rechnen:

PRINT COMPUTER+MAX

Nach dem Drücken der RETURN-Taste steht dann auf dem Bildschirm die Zahl zehn.

Der zweite Spieler, der natürlich nicht gesehen hat, was der erste dem Computer eingegeben hat, versucht nun herauszufinden, welche Zahl zu „MAX" gehört. Das ist nicht so schwierig, wie es zunächst aussieht. Denn der Computer kann ja auch als „Quizmaster" arbeiten.

Wenn man eine Aussage eingibt, meldet er, ob diese Aussage richtig oder falsch ist – soweit es sich dabei um eine Rechnung handelt.

Der Computer meldet „Ø" bei einer falschen, „–1" bei einer richtigen Antwort.

Der zweite Spieler könnte nun einfach raten:

PRINT MAX=5

Der Computer meldet „Ø" (das heißt „falsch"), weil seiner Erinnerung nach der erste Spieler dem Wort MAX eine andere Zahl zugeordnet hat.

Wenn der zweite Spieler zufällig gleich die richtige Zahl getroffen und

PRINT MAX=8

eingetippt hätte, würde der Computer natürlich melden: „–1" („richtig").

Der zweite Spieler könnte jetzt einfach alle Zahlen von null bis zehn durchprobieren und den Computer fragen, ob sie zu dem Namen MAX passen.

Es gibt aber eine elegantere als die Durchprobier-Methode.

Man kann den Computer auch fragen, ob eine Zahl größer (beziehungsweise kleiner) als eine andere ist. Die Zeichen dafür kennst du wahrscheinlich schon aus der Schule:

$\geq$ bedeutet „größer als“

und

$\leq$ bedeutet „kleiner als“. (Merk dir das so: Das „$<$“ ist das unvollständige „k“ vor dem Wort „kleiner“.

Richtig sind zum Beispiel die Aussagen

$2 > 1$ (zwei ist größer als eins)

oder

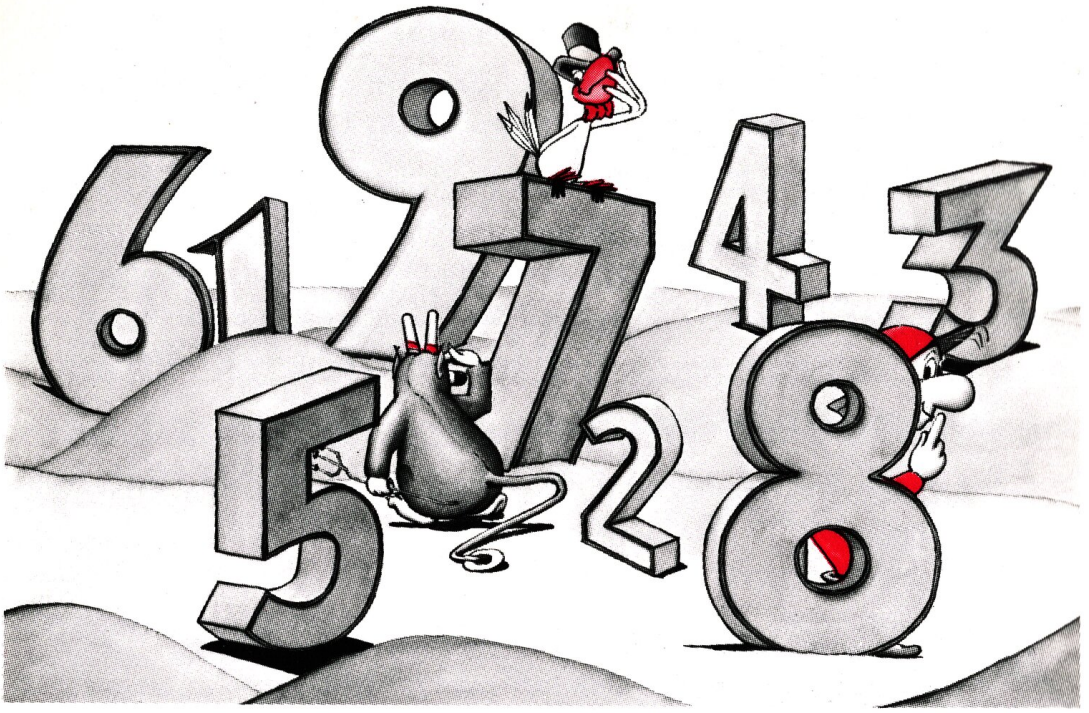
$10 < 20$ (zehn ist kleiner als zwanzig).

(Du kannst deinem Computer zwischendurch ruhig ein paar Größer- und Kleiner-Aussagen vorsetzen und ihn als Quizmaster fragen, ob deine Behauptungen richtig oder falsch sind. Die Zahlen und Namen unseres COMPUTER-MAX-Spiels behält er unterdessen im Gedächtnis.)

Nachdem wir ein bißchen herumprobiert haben, kehren wir zu unserem Spiel zurück.

Der zweite Spieler hat die elegante Methode gefunden und tippt

PRINT MAX $<$ 5



Darauf antwortet der Computer mit einer „Ø“ („falsch“). Denn dem Wort MAX (eigentlich nur MA) hat der erste Spieler ja die Acht zugeordnet. Und acht ist nicht kleiner als fünf.

Nun weiß also der zweite Spieler, daß die Zahl für „MAX“ größer als fünf oder gleich fünf sein muß. Zugleich weiß er auch, daß die Zahl für „COMPUTER“ kleiner oder gleich fünf sein muß: Sonst könnten beide Zahlen zusammen ja nicht zehn ergeben.

Ø; 1; 2; 3; 4; 5; 6; 7; 8; 9; 10

Jetzt muß der zweite Spieler nur noch sechs Zahlen für das Wort „MAX“ in Betracht ziehen (fünf, sechs, sieben, acht, neun und zehn).

Geschickt wie er ist, fragt der zweite Spieler jetzt

PRINT MAX >7

Der Computer antwortet „–1“ („richtig“).

5; 6; 7; 8, 9, 10

Der Max steckt also hinter der Acht, Neun oder Zehn. Der zweite Spieler muß also noch höchstens zweimal probieren; dann hat er mit

PRINT MAX=8

die Aufgabe gelöst.

Überlege einmal, was passiert wäre, wenn man für „MAX" die Fünf und für „COMPUTER" auch die Fünf eingegeben hätte. Oder probier es besser gleich aus.

Man kann statt der Frage, ob eine Zahl größer oder kleiner ist als eine andere, auch dies fragen:

Ist die erste Zahl größer oder gleich groß bzw. kleiner oder gleich groß? Neben den Zeichen für größer und kleiner gibt es noch zwei weitere, die ähnlich aussehen. So kennst du sie vielleicht aus der Schule:

≧ (größer oder gleich)
≦ (kleiner oder gleich)
Der Computer schreibt sie so:
>=
<=

Zwischen zwei Zahlen (oder Variable) geschrieben bedeutet

>= : die linke Zahl ist größer oder gleich der rechten

<= : die linke Zahl ist kleiner oder gleich der rechten.

Die Aussagen

PRINT 5>=5

und

PRINT 5<=5

bewertet der Computer gleich, und zwar mit „–1“; denn beide Aussagen sind richtig.

Die Aussagen

5>5

beziehungsweise

5<5

dagegen sind beide falsch.

Jetzt zeigen wir dir unser **GEHEIMCODE-KNAKKER-SPIEL** für Computer-Geheimagenten.

Vor Spielbeginn geben wir das NEW-Kommando ein, um den Speicher des Computers zu leeren. Sonst könnte er vielleicht durcheinanderkommen.

Der erste Spieler, 001, denkt sich ein Wort, zum Beispiel das Wort COMPUTER. Dann, beginnend mit eins, numeriert er die Buchstaben.

„COMPUTER" verschlüsselt der erste Spieler dann so:

C = 1 (Immer mit „RETURN" bestätigen!)
O = 2
M = 3
P = 4
U = 5
T = 6
E = 7
R = 8

Danach löscht er den Bildschirm. Die Verschlüsselung behält der Computer im Gedächtnis.

Der zweite Computer-Agent, 002, der das Wort entschlüsseln soll, darf den Bildschirm erst jetzt sehen.

Ihm wird nur gesagt, wieviel Buchstaben das gesuchte Wort hat. Für den Anfang verwenden wir nur Wörter, in denen kein Buchstabe zweimal vorkommt.

Nun malt er sich auf einem Zettel so viele Kästchen, wie das Wort Buchstaben hat.

Dann fragt er den Computer, ob ein Buchstabe im Geheimcode-Wort steckt oder nicht, und wenn ja, an welchem Platz. Dazu tippt er einfach:

PRINT A

Der Computer antwortet mit „Ø". Denn den Buchstaben A hat Agent 001 nicht mit einer Zahl gleichge-

setzt, das heißt, nicht verschlüsselt. Solange du ihm nichts anderes angibst, nimmt der Computer für jeden Buchstaben die „Ø“ an.

Dieselbe Antwort bekommt Agent 002 bei der Frage nach dem B. Auf

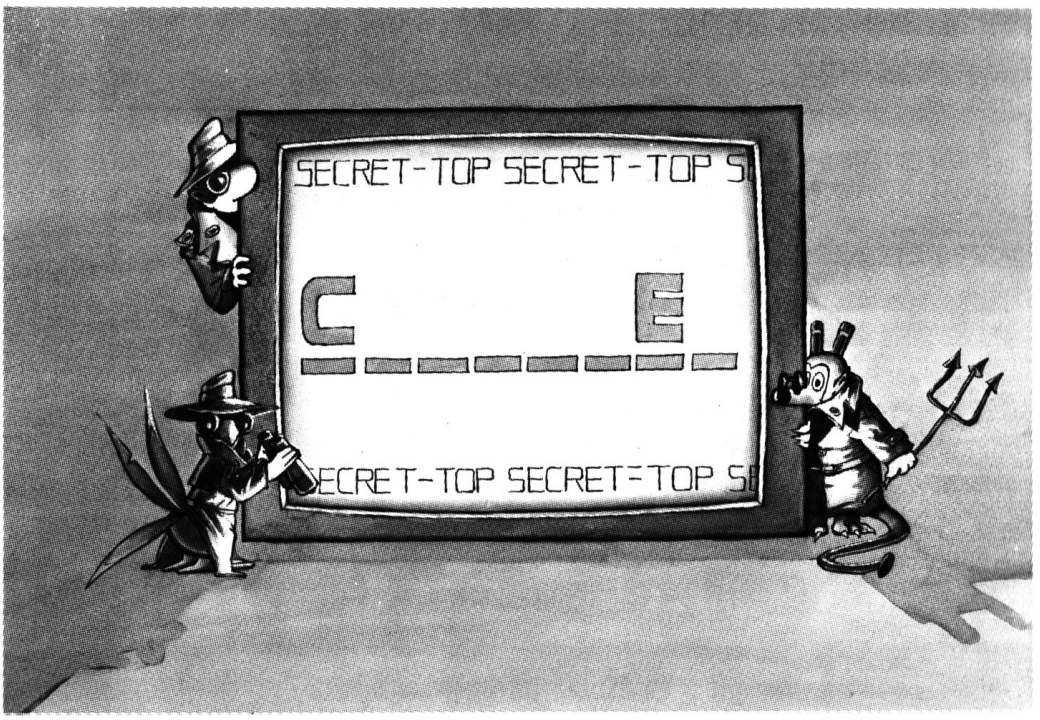
PRINT B

antwortet der Computer ebenfalls mit „Ø“.

Aber auf die Frage nach dem C kommt eine andere Antwort, nämlich die „1“.

Diese „1“ ist nicht zu verwechseln mit der Richtig-Meldung „–1“. Die „1“ sagt, an welcher Stelle im verschlüsselten Wort der Buchstabe C vorkommt, nämlich an der ersten. Der Agent 002 trägt also ins erste Kästchen ein „C“ ein und weiß jetzt, daß das codierte Wort mit „C“ anfängt.

Für „D“ meldet der Computer wieder eine „Ø“, denn der Buchstabe D kommt in dem geheimen Wort nicht vor.



Und bei „E“ erscheint als Antwort des Computers die „7“.

Also ist das „E“ in dem verschlüsselten Wort enthalten, und zwar an siebter Stelle. Das kann der zweite Geheimagent nun eintragen.

So geht es dem Alphabet nach weiter, bis 002 das Wort ganz entschlüsselt hat.

Noch spannender ist es, wenn man nach einigen Treffern versucht, das Wort selbständig zu ergänzen und dabei den Computer wieder als Lügendetektor benutzt.

Wenn der zweite Agent schon bis

```
COMP . . E .
```

gekommen ist, kann er versuchen, die noch fehlenden Buchstaben herauszufinden. Das ist in diesem Beispiel nicht allzu schwer, aber manchmal ist es auch gar nicht so einfach.

002 kombiniert und riskiert dann zu fragen, ob das „U“ im Wort vorkommt:

```
PRINT U
```

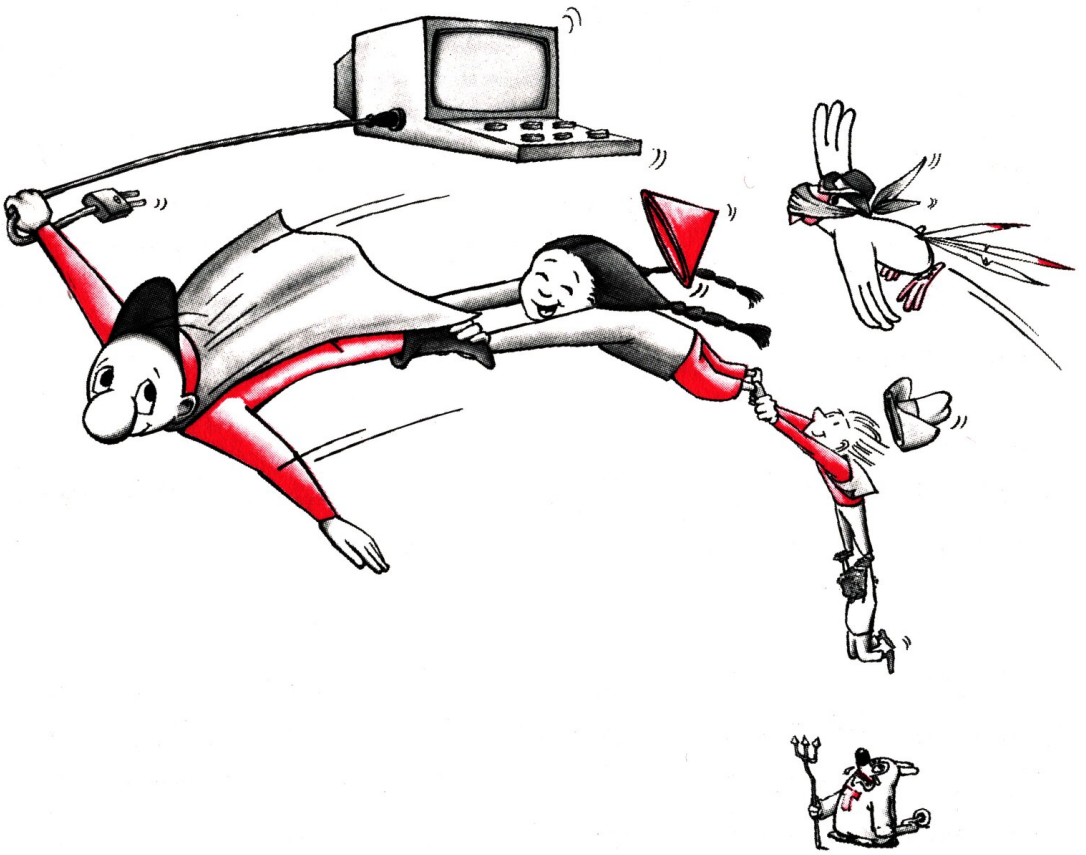
und nach der Antwort „5“ weiß er, daß er richtig getippt hat: Das Geheimwort heißt wohl COMPUTER. Zur Sicherheit fragt er das „T“ und das „R“ auch noch ab.

Der beste Computer-Geheimagent ist, wer so ein verschlüsseltes Wort am schnellsten herausbekommt. Ein bißchen Glück ist natürlich auch dabei.

So, und zum Schluß haben wir noch zwei Spiele, die du von der Diskette laden oder nach unseren Listings von Computer-Profis aus deinem Bekanntenkreis eintippen lassen kannst.

Beide passen sehr gut zu unserem Geheimagen-

ten-Spiel. Mit dem **COMPUTER-MEMORY-SPIEL** kannst du dein Gedächtnis trainieren. Und unser **HUBSCHRAUBER-ABENTEUER-SPIEL** verlangt schnelle Reaktion; du mußt dich ein bißchen auskennen in der Welt (und lernst es durch unser Spiel), und du mußt einer Gefahr ausweichen.



Tips & Tricks für das Eingeben der Listings

Beim Abtippen der Listings schleichen sich gern Fehler ein. Die folgenden Tips sollen dich davor bewahren und helfen, Fehler schnell zu finden.

TIP 1

Lege das Listing, das du eintippen willst, so vor dich hin, daß du es gut lesen kannst. Als Lesehilfe kannst du ein unbeschriebenes Blatt Papier verwenden. Lege es genau unter die Zeile, die du als nächstes eingeben willst.

TIP 2

Hake mit einem Stift jede eingegebene Zeile ab. So gibst du keine Zeile doppelt ein.

TIP 3

Speichere dein Programm des öfteren mal ab, zum Beispiel alle 10 Zeilen. Sollte inzwischen der Rechner abstürzen, gehen nur die Zeilen verloren, die du noch nicht abgespeichert hast.

TIP 4

Spätestens wenn du das eingegebene Programm starten willst, solltest du das Programm zuvor abspeichern. Denn eventuelle Fehler können den Rechner so blockieren, daß du nur durch Ausschalten des Rechners wieder in den normalen Betrieb gelangst, und das Programm wäre damit verloren. Sobald du das Programm abgespeichert hast,

kannst du es wieder in den Rechner laden und den Fehler suchen.

TIP 5

Falls du mit BASIC schon einigermaßen vertraut bist, noch ein weiterer Tip: Wenn Verzweigungen kommen (GOTO, GOSUB), mach's so, wie der Rechner später, „springe" in diese Verzweigungen und tippe diese zuerst ab, bevor du mit dem Hauptprogramm weitermachst. Das hat den Vorteil, daß du jetzt schon Probestarts des Programmes machen, und schon etwaige Fehler ausbessern kannst. Beachte dabei TIP 4.

Zum Go-Spielen verwendet man eine Fläche von 19 x 19 Stellen, die man (vielleicht mit den Zahlen) am besten erst einmal auf den Bildschirm "zeichnet".

Abwechselnd darf der Spieler mit dem weißen und der Spieler mit dem schwarzen Kreis einen „Stein" setzen.

Es kommt darauf an, möglichst lange „Ketten" mit seinen Steinen zu bilden. Die Stellen, die innerhalb einer geschlossenen Kette liegen, werden dem jeweiligen Spieler als Punkte gutgeschrieben.

Und wenn man gegnerische Steine (wie im unteren Beispiel) mit seiner Kette einschließt, dann gehören sie dem, der sie eingeschlossen hat.

Es gibt übrigens sehr viele verschiedene Variationen der Spielregeln für Go und Go Ban(g). Man kann also auch seine eigenen Regeln entwickeln.

Eine häufig angewendete GO-BANG-Regel: Wer zuerst 4 „Steine" in einer Reihe fertigbringt, hat gewonnen. Die Reihe kann senkrecht, waagrecht oder

schräg verlaufen und darf natürlich nicht abbiegen, d. h. um die Ecke gehen. Einfacher ist das Spiel, wenn es nur drei in einer Reihe sein müssen. Sollen mehr Steine in einer Reihe liegen, etwa 5, dann wird es besonders spannend – aber auch knifflig.

Listings der Spiele von Band 1

1. Schlangenspiel
2. Grand Prix

2a. Programmbausteine von „Grand Prix“ (Module)

3. Rechtschreibspiel
4. Memory
5. Heli-Abenteuer

```
100 :       ---------------
101 :        STEUERZEICHEN
102 :       ---------------
103 :
104 :     1) CURSORSTEUERTASTEN
105 :    "▶" CRSR            CURSOR RECHTS
106 :    "◀" SHIFT CRSR      CURSOR LINKS
107 :    "▼" CRSR            CURSOR RUNTER
108 :    "▲" SHIFT CRSR      CURSOR HOCH
109 :    "▪" CLRHOME         CURSOR INS LINKE ECK
110 :    "▪" SHIFT CLRHOME   BILDSCHIRM LOESCHEN
111 :
112 :     2) FUNKTIONSTASTEN
113 :    "■" F1              FUNKTIONSTASTE 1
114 :    "■" SHIFT F1        FUNKTIONSTASTE 2
115 :    "■" F3              FUNKTIONSTASTE 3
116 :    "■" SHIFT F3        FUNKTIONSTASTE 4
117 :    "■" F5              FUNKTIONSTASTE 5
118 :    "■" SHIFT F5        FUNKTIONSTASTE 6
119 :    "■" F7              FUNKTIONSTASTE 7
120 :    "■" SHIFT F7        FUNKTIONSTASTE 8
121 :
122 :     3) FARBEN
```

```
123 :    "■" CTRL 1          SCHWARZ
124 :    "■" CTRL 2          WEISS
125 :    "■" CTRL 3          ROT
126 :    "■" CTRL 4          TUERKIS
127 :    "■" CTRL 5          VIOLETT
128 :    "■" CTRL 6          GRUEN
129 :    "■" CTRL 7          BLAU
130 :    "■" CTRL 8          GELB
131 :    "■" C= 1            ORANGE
132 :    "■" C= 2            BRAUN
133 :    "■" C= 3            HELLROT
134 :    "■" C= 4            GRAU 1
135 :    "■" C= 5            GRAU 2
136 :    "■" C= 6            HELLGRUEN
137 :    "■" C= 7            HELLBLAU
138 :    "■" C= 8            GRAU 3
139 :
140 :     4) SONSTIGE STEUERZEICHEN
141 :    "■" CTRL 9          REVERSSCHRIFT AN
142 :    "■" CTRL 0          REVERSSCHRIFT AUS

READY.
```

Was heißt was in BASIC

Die wichtigsten BASIC-Begriffe aus diesem Buch

So heißt es in BASIC:	**Und das soll der Computer tun:**
AND	Lasse bei IF . . . THEN mehrere Bedingungen gelten.

Beispiel

```
10 INPUT A
20 INPUT B
30 PRINT A+B
40 A=A+1
50 B=B+2
60 IF A>10 AND B>20THEN END
70 GOTO 30
```

END	Beende das Programm
GOTO	Gehe zur Zeile deren Nummer nach GOTO kommt

Beispiel:

```
10 PRINT "HALLO"
20 PRINT "COMPUTER"
30 GOTO 10
```

IF . . . THEN	WENN eine Bedingung erfüllt ist, DANN springe zur Zeile, die nach THEN steht

Beispiel:

```
10 INPUT A
20 IF A>10 THEN GOTO 10
30 IF A<10 THEN PRINT "NA ENDLICH"
40 END
```

INPUT

Frage erst nach einer Eingabe (zum Beispiel einer Zahl oder einem Buchstaben), bevor du ein Programm weiter bearbeitest.

Beispiel:

```
10 INPUT A
20 PRINT "A*10=";A*10
30 GOTO 10
40 END
```

LIST

Zeige das Programm-Listing auf dem Bildschirm.

LOAD

Lade ein Programm vom Datenträger (Kassette oder Diskette)

Beispiel:

LOAD „Leiterspiel“ lädt ein Programm mit dem Namen „Leiterspiel“ von der Kassette. Auf der Datasette oder auf dem Kassettenrecorder wird PLAY gedrückt.

LOAD "Leiterspiel", 8 lädt das Programm „Leiterspiel" von der Diskettenstation.

NEW — Lösche dein Gedächtnis.

OR — Lasse bei IF . . . THEN mehrere Bedingungen gelten.

Beispiel:

```
10 INPUT A
20 INPUT B
30 PRINT A+B
40 A=A+1
50 B=B+2
60 IF A=10 OR B>20THEN END
70 GOTO 30
```

PRINT — mit Anführungszeichen: Schreibe oder zeichne auf den Bildschirm, was zwischen den Anführungszeichen steht.

Beispiel:

```
10 PRINT"MAX"
```

PRINT — ohne Anführungszeichen: Führe aus, rechne aus, was hinter dem PRINT steht.

Beispiel:

```
10 PRINT 23-9
```

REM — Schreibe eine Bemerkung des Programmierers in das

Listing. Sie soll das Programm selbst nicht beeinflussen.

Beispiel:

```
10 REM *ZWEIER-EINMALEINS*
20 INPUT A
30 PRINT A*2
40 GOTO 20
```

RETURN

Funktionstaste:
Merke dir, was ich gerade geschrieben habe und gehe mit dem Cursor an den Anfang der nächsten Zeile.

RUN

Führe das Programm aus.

RUN/STOP

Funktionstaste:
Halte das Programm dort an, wo du gerade bist und melde mir mit BREAK IN . . . (Zeilennummer) wo die Unterbrechung stattgefunden hat.

SAVE

Speichere ein Programm auf dem Datenträger (Kassette oder Diskette) ab.

Beispiel:

SAVE "Leiterspiel" speichert ein Programm auf Kassette. Es müssen die beiden Tasten PLAY und RECORD gedrückt werden.

Save "Leiterspiel", 8 speichert das Programm „Leiterspiel" auf Diskette.

Und die Steuerzeichen:

"▶" Cursor um eine Stelle nach rechts.

"◀" Cursor um eine Stelle nach links.

"▲" Cursor um eine Zeile nach oben.

"▼" Cursor um eine Zeile nach unten.

"●" Bildschirm löschen.

„Die in diesen Listings enthaltenen Programme sind aus Bausteinen zusammengesetzt. Einige dieser Bausteine sind zum einwandfreien Funktionieren nicht unbedingt nötig, sondern dienen nur als „Bild" bzw. „Grafik". Jeder Programmbaustein ist durch REM-Zeilen gekennzeichnet. Beim Abtippen kann man sich also schon überlegen, ob es nicht einfacher ist, auf die entbehrlichen Programmbausteine zu verzichten! Das spart Zeit, und man hat auch weniger Tippfehler zu befürchten."

```
100 REM ------------------
110 REM # SCHLANGENSPIEL #
120 REM ------------------
130 PRINT"□■":POKE53280,6:POKE53281,6:POKE788,49:POKE792,71
140 INPUT"WIE HEISST DU";NA$
150 IM=3:GOSUB5500
160 PRINT"WOLLEN WIR EIN KLEINES SPIEL MACHEN,    "NA$"?"
170 IM=1:GOSUB5500
180 GOSUB5000
190 ONAGOTO1000,200
200 PRINT"@ICH MOECHTE ABER SO GERNE,":PRINTNA$
210 PRINT"@WILLST DU VIELLEICHT NICHT DOCH NOCH?"
230 IM=1:GOSUB5500
240 GOSUB5000
250 ONAGOTO1000,300
260 :
300 IM=1:GOSUB5500
310 PRINT"VIELLEICHT EIN ANDERES MAL, "NA$
320 IM=1:GOSUB5500
330 PRINT"       TSCHUESS !!!"
340 GOTO2080
350 :
1000 REM #### SPIELROUTINE ####
1010 PRINT"□"
1020 PRINT"ICH SCHREIBE DIR EINE REIHE VON >X< UND >-< AUF DEN BILDSCHIRM,"
1030 IM=1:GOSUB5500
1040 PRINT"ZUM BEISPIEL SO:"
1050 PRINT"-X-XX---XXX-XX-X-X"
1060 IM=1:GOSUB5500
```

```
1070 PRINT"UND DU KANNST DANN DIESE REIHE NACHTIP- PEN, "NA$
1080 IM=2:GOSUB5500
1090 PRINT"BITTE GIB MIR NUN EINE ZAHL ZWISCHEN    1 UND 20 EIN."
1100 IM=1:GOSUB5500
1110 PRINT"JE GROESSER DIE ZAHL IST, DESTO SCHWIE- RIGER WIRD'S"
1120 INPUTA
1130 IM=1:GOSUB5500
1140 IFA<1ORA>20 THENPRINT"ICH SAGTE ZWISCHEN 1 UND 20":GOTO1080
1150 D$=LEFT$("X-XX-X-XXX---X-X--X",A)
1160 PRINT"␣"
1170 PRINT"DIE REIHE LAUTET:"
1180 PRINTD$
1190 IM=3:GOSUB5500
1200 PRINT"JETZT KANNST DU DIE REIHE NACHTIPPEN,   "NA$
1210 IM=2:GOSUB5500
1220 INPUTE$
1230 IM=3:GOSUB5500
1240 IFD$=E$THEN1300
1250 PRINT"DAS WAR LEIDER NICHT GANZ RICHTIG,      "NA$
1260 IM=1:GOSUB5500
1270 PRINT"VERSUCH ES NOCH EINMAL !"
1280 FORI=1TO3000:NEXT
1290 GOTO1160
1300 PRINT"␣"
1310 PRINT"STIMMT GENAU, "NA$
1320 FORI=1TO3000:NEXT
1330 GOSUB10000
1340 FORI=1TO10000:NEXT
1350 :
```

```
2000 REM #### SPIELENDE ? ####
2010 PRINT"□":POKE53280,6:POKE53281,6
2020 PRINT"MOECHTEST DU WEITERMACHEN, "NA$" ?"
2030 IM=2:GOSUB5500
2040 GOSUB5000
2050 ONAGOTO1000,2060
2060 PRINT"□"
2070 PRINT"BIS ZUM NAECHSTEN MAL, "NA$
2080 FORI=1TO5000:NEXT
2090 PRINT"□"
2100 END
2110 :
5000 REM #### ABFRAGEROUTINE ####
5010 PRINT"GEBE >JA< ODER >NEIN< EIN!□":INPUTA$
5020 IM=1:GOSUB5500
5030 IFA$="JA"THENA=1:RETURN
5040 IFA$="NEIN"THENA=2:RETURN
5050 PRINT:PRINT"DIESE ANTWORT VERSTEHE ICH NICHT !"
5060 PRINT"VERSUCHE ES DOCH BITTE NOCH EINMAL,     "NA$"□□"
5070 FORI=1TO5000:NEXT
5080 GOTO5000
5090 :
5500 REM #### VERZOEGERUNGSROUTINE ####
5510 FORI=1TOIM
5520 PRINT
5530 FORJ=1TO500:NEXTJ
5540 NEXTI
5550 RETURN
5560 :
```

```
10000 REM #### BELOHNUNGSROUTINE ####
10010 POKE53280,1:POKE53281,1:PRINT"■";
10020 PRINT"■
10030 PRINT"
10040 PRINT"
10050 PRINT"■
10060 PRINT"■
10070 PRINT"■
10080 PRINT"■
10090 PRINT"■
10100 PRINT"■
10110 PRINT"■
10120 PRINT"
10130 PRINT"
10140 PRINT"
10150 PRINT"
10160 PRINT"
10170 PRINT"
10180 PRINT"
10190 PRINT"
10200 PRINT"
10210 PRINT"
10220 PRINT"
10230 RETURN

READY.
```

```
100 REM ----------------
110 REM >> GRAND PRIX <<
120 REM ----------------
130 DIMB$(10):POKE649,1
140 FORI=0TO9
150 READS$(I)
160 NEXT:POKE788,49:POKE792,71
170 REM -----------------
180 REM > HAUPTPROGRAMM <
190 REM -----------------
200 GOSUB2000
210 GOSUB3000
220 GOSUB5000
230 GOTO200
240 REM --------------
250 REM > DATAZEILEN <
260 REM --------------
270 DATA"   10.0  NIKI LAUDA"
280 DATA"   15.0  ALAIN PROST"
290 DATA"   20.0  ELIO DE ANGELIS"
300 DATA"   25.0  MICHELE ALBORETO"
310 DATA"   50.0  NELSON PIQUET"
320 DATA"   75.0  RENE ARNOUX"
330 DATA"  100.0  KEKE ROSBERG"
340 DATA"  250.0  RICCARDO PATRESE"
350 DATA"  500.0  MANFRED WINKELHOCK"
360 DATA"  750.0  MAX SURER"
2000 REM ---------------
2010 REM ## TITELBILD ##
```

```
2020 REM ---------------
2030 PRINT"■■"CHR$(142);:POKE53281,11:POKE53280,0
2040 PRINT"■■                                                    ";
2050 PRINT"                                                      ";
2060 PRINT"                                                      ";
2070 PRINT"                                                      ";
2080 PRINT"                                                      ";
2090 PRINT"                                                      ";
2100 PRINT"                                                        ";
2110 PRINT"                                                          ";
2120 PRINT"                                                          ";
2130 PRINT"                                                      ";
2140 PRINT"                                                      ";
2150 PRINT"■                                                     ";
2160 PRINT"                                                      ";
2170 PRINT"                                                      ";
2180 PRINT"                                                      ";
2190 PRINT"                                                      ";
2200 PRINT"                                                      ";
2210 PRINT"
2220 PRINT"
2230 PRINT"
2240 PRINT"
2250 PRINT"
2280 I=.125
2290 I=I*2
2300 PRINT"■■■
```

```
2330 PRINT"■■■■■■■■■■■■■■■■■■■■ ■■■■■■■■■■■■■■■■■■■
```

```
3180 PRINT"OKAY!":NEXTI
3190 FORI=1TO2
3200 A=INT(RND(1)*10+1):B=INT(RND(1)*10+1)
3210 PRINT"@"A"*"B
3220 INPUTC
3230 IFC<>A*BTHENPRINT"FALSCH! DAS ERGEBNIS LAUTET"A*B:GOTO3200
3240 PRINT"OKAY!":NEXTI
3250 FORI=1TO2
3260 A=INT(RND(1)*10+1):B=INT(RND(1)*10+1):IFA<BTHENC=A:A=B:B=C
3270 PRINT"@"A"-"B
3280 INPUTC
3290 IFC<>A-BTHENPRINT"FALSCH! DAS ERGEBNIS LAUTET"A-B:GOTO3260
3300 PRINT"OKAY!":NEXTI
3310 FORI=1TO2
3320 A=INT(RND(1)*10+1):B=INT(RND(1)*10+1)*A
3330 PRINT"@"B":"A
3340 INPUTC
3350 IFC<>B/ATHENPRINT"FALSCH! DAS ERGEBNIS LAUTET"B/A:GOTO3320
3360 PRINT"OKAY!":NEXTI
3370 FORI=0TO2500:NEXT:PRINT"␣BRAVO!!"
3380 PRINT"@NUN HAST DU DEINE FAHRERLIZENZ."
3390 FORI=0TO4000:NEXT:RETURN
5000 REM ------------
5010 REM ## RENNEN ##
5020 REM ------------
```

```
5030 A$(0)="  ▒▒▒
```

```
5330 POKEC,81:POKEC1,102
5340 C1=C:GOTO5250
5350 IFPEEK(C)=93THEN5440
5360 POKE53280,1:POKE53280,0
5370 PRINT"■QQQQQQQQQQQQQ■■■
```

```
100 REM ----------------------
110 REM ## RECHTSCHREIBUNG ##
120 REM ----------------------
130 POKE53280,6:POKE53281,6:PRINT"■";:POKE788,49:POKE792,71
140 DIMB(26),C(11,2)
150 SP$="                                                  ":B1=0
160 FORH=0TO11:READA$:NEXT
170 FORH=0TO2
180 READA$:GOSUB2090
190 NEXT
200 GETA$:IFA$="N"THEN250
210 IFA$<>"J"THEN200
220 PRINT"■■";
230 FORH=0TO8:READA$:GOSUB2090:NEXT
240 GETA$:IFA$=""THEN240
250 RESTORE
260 PRINT"■■■
```

```
510 B=1
520 IFB<1THENB=B1
530 IFB>B1THENB=1
540 POKE1064+B2,96:B2=B(B)
550 POKE1064+B(B),30:POKE55336+B(B),0
560 GETA$:IFA$=""THEN560
570 IFA$="■"THENB=B+1:GOTO530
580 IFA$="■"THENB=B-1:GOTO520
590 IFA$="■"THENPOKE1064+B2,96:GOTO1000
600 IFA$<"A"ORA$>"Z"THEN560
610 A=ASC(A$)-64
620 POKE1024+B(B),A
630 B=B+1:GOTO520
1000 REM ## KORREKTUR ##
1010 RU=RU+1
1020 RESTORE:FE=0
1030 PRINT"■■■
```

```
1160 A$="ES SIND NOCH"+STR$(FE)+" LUECKE(N) AUSZUFUELLEN":GOSUB2090
1170 FORI=0TO2000:NEXT
1180 POKE53280,6:POKE53281,6
1190 PRINT"■■■
```

```
1615 A$="HALTEN!":GOSUB2090:RETURN
1620 A$="VIEL BESSER ALS EIN ANFAENGER BIST DU":GOSUB2090
1630 A$="NICHT!":GOSUB2090:RETURN
1640 A$="ICH SAGE DIR LIEBER NICHT, ALS WAS ICH":GOSUB2090
1650 A$="DICH EINSCHAETZE!":GOSUB2090
1660 A$=SP$:GOSUB2090
1670 A$="DU BRAUCHTEST MEHR ALS 4 RUNDEN!":GOSUB2090
1680 REM ## ENDE ##
1690 A$="WENN DU NOCH MAL SPIELEN WILLST, DANN":GOSUB2090
1700 A$="TIPPE RUN EIN UND DRUECKE DIE RETURN-":GOSUB2090
1710 A$="TASTE":GOSUB2090
1720 END
2000 REM ## LUECKENMACHER ##
2010 IFA$=""THENPRINT"@":RETURN
2020 A=LEN(A$):IFA<5THEN2090
2030 A=A-2:FORI=1TO2
2040 A1=INT(RND(1)*A+2)
2050 IFMID$(A$,A1,1)<"A"ORMID$(A$,A1,1)>"Z"THEN2040
2060 A$=LEFT$(A$,A1-1)+"▒"+MID$(A$,A1+1)
2070 B1=B1+1:B(B1)=A1-1+H*80:C(H,I)=A1
2080 NEXT
2090 A$=LEFT$(A$+SP$,40)
2100 FORI=1TO40
2110 PRINTMID$(A$,I,1);
2120 FORJ=0TO1.00
2130 NEXTJ,I:PRINT:RETURN
2200 REM ## KORREKTURUNTERPROGRAMM ##
2210 A$=LEFT$(A$+SP$,40)
2220 FORI=1TO40
```

```
2230 A1$=MID$(A$,I,1)
2240 IFA1$<"A"OR(A1$)"Z"ANDA1$<>"Ä")THEN2330
2250 B=PEEK(1023+I+H*80)
2260 IFA1$<>CHR$(B+64)THENFE=FE+1:PRINT"■"A1$"■";:GOTO2340
2270 FORK=1TO2
2280 IFC(H,K)=ITHENC(H,K)=0
2290 NEXTK
2300 FORK=1TOB1
2310 IFB(K)=H*80+I-1THENB(K)=0
2320 NEXTK
2330 PRINTA1$;
2340 NEXTI:PRINT:RETURN
3000 REM ## TEXT ##
3010 DATA"EINE WAHRE GESCHICHTE"
3020 DATA
3030 DATA"IN ARGENTINIEN WURDEN IM JAHR 1978 IN"
3040 DATA"EINER KLEINEN STADT DREI LEHRER ABGE-"
3050 DATA"SETZT. BEI EINER KONTROLLE STELLTE SICH"
3060 DATA"HERAUS, DASS DIE DREI LEHRER WEDER LE-"
3070 DATA"SEN NOCH SCHREIBEN KONNTEN. IM UNTER-"
3080 DATA"RICHT UNTERHIELTEN SIE IHRE SCHUELER,"
3090 DATA"INDEM SIE IHNEN VIELE, VIELE GESCHICH-"
3100 DATA"TEN ERZAEHLTEN ..."
3110 DATA
3120 DATA
3130 DATA"   >  > >> RECHTSCHREIBSPIEL << <  <"
3140 DATA
3150 DATA"BRAUCHST DU DIE ANLEITUNG (J/N)?"
3160 DATA"IN DIESEM SPIEL GEHT ES DARUM, EINEN"
```

```
3170 DATA"DURCH DEN COMPUTER ENTSTANDENEN LUECKEN-"
3180 DATA"TEXT AUSZUFUELLEN. MAN KANN DABEI IMMER
3190 DATA"NUR DIE LUECKE BEARBEITEN, UNTER DER DER
3200 DATA"PFEIL STEHT. DEN PFEIL KANN MAN MIT DEN
3210 DATA"CURSORSTEUERTASTEN BEWEGEN. ES SIND NUR
3220 DATA"BUCHSTABEN ZU ERGAENZEN.
3230 DATA"
3240 DATA"FERTIG? DRUECKE BITTE IRGENDEINE TASTE!"
READY.
```

MEMORY

Memory ist ein Spiel, bei dem du dein Gedächtnis trainieren kannst. Dein Spielpartner ist der Computer.

Wie im Memory-Spiel mit Karten soll man sich merken, welche Zeichen auf den verdeckten „Karten" liegen, und herausfinden, auf welchen Plätzen gleiche Karten liegen.

Jeder Spieler darf nacheinander zwei „Karten" aufdekken. Hat er dabei ein Pärchen gefunden, ist er noch einmal dran, und er bekommt einen Punkt. Wenn ein Pärchen gefunden worden ist, bleiben die Zeichen offen, ansonsten werden sie wieder verdeckt. Gewonnen hat, wer die meisten Pärchen gesammelt hat, nachdem alle Zeichen aufgedeckt sind.

Der Spieler hat die Wahl zwischen 16, 32, 48 und – für echte Memory-Profis – 64 Feldern mit Zeichen-Paaren. Zusätzlich kann er die Gedächtnisstärke seines Gegenspielers C64 bestimmen. Die verdeckten Zeichen sind bei jedem Spiel wieder neu auf dem Feld verteilt.

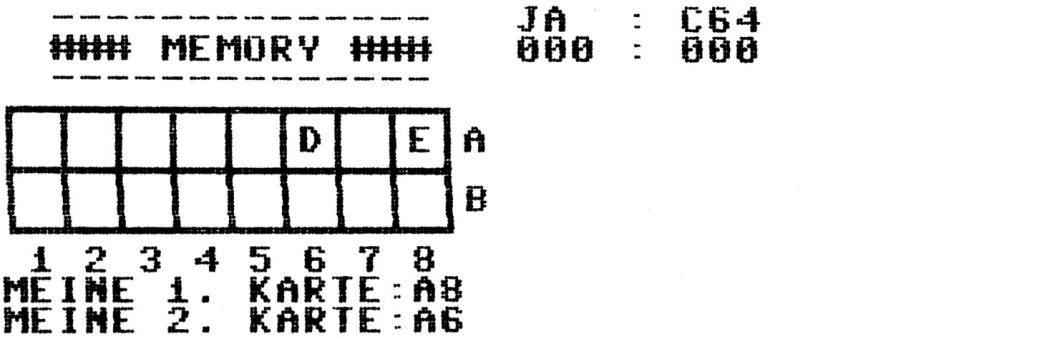
```
 ---------------
 ### MEMORY ###
 ---------------

HALLO !!
HAST DU LUST MIT MIR [illegible] ZU SPIELEN?
(JA ODER NEIN)
DEINE ANFANGSBUCHSTABEN? PF
SPIELFELDGROESSE:
[illegible] 16 FELDER
[illegible] 32 FELDER
[illegible] 48 FELDER
[illegible] 64 FELDER
DRUECKE GEWUENSCHTE ZAHL.
```

Nach dem Laden des Spiels fragt der Computer dich, ob du mit ihm ein Spiel machen willst. Tippe J für Ja oder N für Nein ein. Nun fragt er mit „Deine Anfangsbuchstaben?" nach deinem Namen. Du kannst ihm drei Buchstaben als Abkürzung für deinen Namen eintippen.

Als nächstes wählst du die Spielfeldgröße und das Gedächtnis des Computers. Schließlich fragt der Com-

puter noch: „Wer soll anfangen?“ Antworte bitte mit S für Spieler oder C für Computer. Bei allen diesen Eingaben brauchst du übrigens nicht mit RETURN zu bestätigen.

Sobald du beginnst, fragt der Computer nach der ersten Karte. Du gibst die Koordinaten eines Feldes ein. Das heißt: der Buchstabe bestimmt die waagrechte und die Zahl die senkrechte Reihe.



Man gibt immer zuerst den Buchstaben und dann die Zahl ein. Nach einer falschen Eingabe springt der Cursor wieder an den Anfang der Eingabe zurück. Du kannst deine Eingaben auch mit DEL „radieren“, solange sie noch nicht mit RETURN bestätigt sind. Rechts oben zählt der Computer deine und seine Punkte.

```
100 REM --------------
110 REM #### MEMORY ####
120 REM --------------
130 POKE53280,6:POKE53281,6:POKE788,49:POKE792,71
140 DIMK$(64),L$(64),C(25)
150 :
1000 REM -------------------
1010 REM #### SPIELANFANG ####
1020 REM -------------------
1030 PRINT"■■"CHR$(142)
1040 PRINT" --------------"
1050 PRINT" #### MEMORY ####"
1060 PRINT" --------------"
1070 PRINT
1080 PRINT"HALLO !!"
1090 PRINT"■HAST DU LUST, MIT MIR ■MEMORY■ ZU SPIELEN? (■J■A ODER ■N■EIN)"
1100 GETA$
1110 IFA$="J"THEN1140
1120 IFA$="N"THENPRINT"■SCHADE!";:END
1130 GOTO1100
1140 PRINT
1150 INPUT"DEINE ANFANGSBUCHSTABEN";NA$
1160 NA$=LEFT$(NA$+"   ",3)
1170 :
1200 REM #### SPIELFELDGROESSE ####
1210 PRINT"■SPIELFELDGROESSE:■"
1220 PRINT"■ 1 ■ 16 FELDER"
1230 PRINT"■ 2 ■ 32 FELDER"
1240 PRINT"■ 3 ■ 48 FELDER"
```

```
1250 PRINT"■ 4 ■ 64 FELDER"
1260 PRINT"■DRUECKE GEWUENSCHTE ZAHL."
1270 GETA$:IFA$<"1"ORA$>"4"THEN1270
1280 SF=VAL(A$)*16
1290 PRINT"■COMPUTERGEDAECHTNIS:■"
1300 A$(1)="■ 1 ■ KEIN GEDAECHTNIS"
1310 A$(2)="■ 2 ■ SCHWACHES GEDAECHTNIS"
1320 A$(3)="■ 3 ■ NORMALES GEDAECHTNIS"
1330 A$(4)="■ 4 ■ GUTES GEDAECHTNIS"
1340 A$(5)="■ 5 ■ SEHR GUTES GEDAECHTNIS"
1350 A$(6)="■ 6 ■ SUPER-GEDAECHTNIS"
1360 IM=(SF/2+1)/4
1370 FORI=1TOIM
1380 PRINTA$(I)
1390 NEXT
1400 PRINT"■DRUECKE GEWUENSCHTE ZAHL."
1410 GETA$:IFVAL(A$)<1ORVAL(A$)>IMTHEN1410
1420 CG=4*(VAL(A$)-1)
1430 :
1500 REM #### KARTEN MISCHEN ####
1510 FORI=1TOSF:K$(I)=" ":NEXT
1520 FORI=0TO1:RESTORE
1530 FORJ=1TOSF/2:READL$(I*SF/2+J):NEXTJ,I
1540 FORI=1TOSF:J=INT(RND(1)*SF+1)
1550 A$=L$(I):L$(I)=L$(J):L$(J)=A$
1560 NEXT
1570 PRINT"■WER SOLL ANFANGEN?"
1580 GETA$:IFA$="S"THEN2010
1590 IFA$="C"THEN3010
```

```
1600 GOTO1580
1610 :
2000 REM ------------------
2010 REM #### SPIELERZUG ####
2020 REM ------------------
2030 IFNA+CO=SF/2THENRUN
2040 GOSUB5030
2050 GOSUB5200:K1$=KA$:K1=K
2060 PRINT"1. KARTE: "K1$
2070 GOSUB5200:K2$=KA$:K2=K
2080 PRINT"1. KARTE: "K1$
2090 PRINT"2. KARTE: "K2$
2100 FORI=0TO6000:NEXT
2110 IFL$(K1)=L$(K2)THENK$(K1)=L$(K1):K$(K2)=L$(K2)
2120 IFK$(K1)=K$(K2)THENNA=NA+1:GOTO2010
2130 K$(K1)=" ":K$(K2)=" ":C(CG+3)=K1:C(CG+4)=K2
2140 :
3000 REM --------------------
3010 REM #### COMPUTERZUG ####
3020 REM --------------------
3030 IFNA+CO=SF/2THENRUN
3040 REM #### COMPUTERGEDAECHTNIS ####
3050 FORI=1TOCG+4
3060 IFK$(C(I))<>" "THENC(I)=0
3070 NEXT
3080 FORI=1TOCG+4
3090 FORJ=I+1TOCG+4
3100 IFC(I)=C(J)THENC(I)=0
3110 NEXTJ,I
```

```
3120 J=0:FORI=1TOCG+4
3130 C=C(I):C(I)=0:C(I-J)=C:IFC(I-J)=0THENJ=J+1
3140 NEXTI
3150 J=4-J:IFJ<1THEN3200
3160 FORI=JTOCG+4
3170 C(I-J)=C(I):NEXT
3200 REM #### COMPUTERZUG ####
3210 C1=0:C2=0:FORI=CG+3TO1STEP-1
3220 FORJ=I+1TOCG+4
3230 IFC(I)=0ORC(J)=0THEN3250
3240 IFL$(C(I))=L$(C(J))THENC1=C(I):C2=C(J)
3250 NEXTJ,I
3260 IFC1>0THEN3280
3270 GOSUB5300:C1=C0
3280 C1$=CHR$(65+INT((C1-1)/8))+RIGHT$(STR$(C1-INT((C1-1)/8)*8),1)
3290 K$(C1)="■"+L$(C1)+"□"
3300 GOSUB5030
3310 PRINT"MEINE 1. KARTE:";C1$
3320 FORI=1TO2500:NEXTI
3330 I=0:IFC2>0THEN3380
3340 I=I+1
3350 IFL$(C(I))=L$(C1)THENC2=C(I):GOTO3380
3360 IFI<CG+4THEN3340
3370 GOSUB5300:C2=C0
3380 C2$=CHR$(65+INT((C2-1)/8))+RIGHT$(STR$(C2-INT((C2-1)/8)*8),1)
3390 K$(C2)="■"+L$(C2)+"□"
3400 GOSUB5030
3410 PRINT"MEINE 1. KARTE:";C1$
3420 PRINT"MEINE 2. KARTE:";C2$
```

```
3430 FORI=0TO8000:NEXT
3440 C(CG+1)=C1:C(CG+2)=C2
3450 IFL$(C1)=L$(C2)THENK$(C1)=L$(C1):K$(C2)=L$(C2)
3460 IFL$(C1)=L$(C2)THENCO=CO+1:GOSUB5030:GOTO3010
3470 K$(C1)=" ":K$(C2)=" "
3480 GOTO2010
5000 REM ------------------
5010 REM ) UNTERPROGRAMME (
5020 REM ------------------
5030 REM ##### SPIELFELD #####
5040 PRINT"■  ---------------   "LEFT$(NA$,3)" : C64"
5050 PRINT"  ##### MEMORY #####   ";
5060 PRINTRIGHT$(STR$(1000+NA),3)" : "RIGHT$(STR$(1000+CO),3)
5070 PRINT"  ---------------"
5080 PRINT"┌─┬─┬─┬─┬─┬─┬─┬─┐
5090 FORI=0TO5F-8STEP8
5100 PRINT"│"K$(I+1)"│"K$(I+2)"│"K$(I+3)"│"K$(I+4)"│";
5110 PRINTK$(I+5)"│"K$(I+6)"│"K$(I+7)"│"K$(I+8)"│"CHR$(I/8+65)
5120 PRINT"├─┼─┼─┼─┼─┼─┼─┼─┤"
5130 NEXT
5140 PRINT"■└─┴─┴─┴─┴─┴─┴─┴─┘"
5150 PRINT" 1 2 3 4 5 6 7 8"
5160 RETURN
5200 REM ##### ZUGEINGABE #####
5210 INPUT"KARTE";KA$
5220 K=VAL(RIGHT$(KA$,1))+(ASC(KA$)-65)*8
5230 IFK<10RK>5FTHENPRINT"■■":GOTO5210
5240 IFK$(K)<>" "THENPRINT"■■":GOTO5210
5250 K$(K)="▦"+L$(K)+"■"
```

```
5260 GOSUB5030:RETURN
5300 REM #### ZUGUEBERLEGUNG ####
5310 C0=INT(RND(1)*SF+1)
5320 C0=C0-1:IFC0<1THENC0=SF
5330 IFK$(C0)<>" "THEN5320
5340 J=0
5350 J=J+1
5360 IFC(J)=C0THEN5320
5370 IFJ<CG+4THEN5350
5380 RETURN
5400 REM #### MEMORYZEICHEN ####
5410 DATAA,B,C,D,E,F,G,H
5420 DATA←,£,@,*,↑,=,!,π
5430 DATA"●","○","♣","♥","♦","♀",+,-
5440 DATA#,$,%,&,(,[,<,?

READY.
```

```
1 REM  SCHNEIDER-----BUCH
2 REM  --------SOFTWARE--
3 REM  ***PRAESENTIERT***
4 REM   * DAS   SPIEL *
5 REM  " =====    HELI-
6 REM  "  ,I-, ADVENTURE
7 REM  "  '--'     VON
8 REM  "   /\
9 REM  >PETER  FRITZSCHE<
10 REM  ######################
11 REM MITARB.:G.SEESSLEN
12 REM  <>COMMODORE 64<>
13 REM ------------------
160 REM ########################
170 POKE53280,0:POKE53281,0:PRINT"■"CHR$(142);:POKE788,49:POKE792,71
180 DIMA$(18)
190 A$(1)="■                                                    "
200 A$(2)="                                                     "
210 A$(3)="                                                     "
220 A$(4)="                                                     "
230 A$(5)="                                                     "
240 A$(6)="■            ** ADVENTURE **                         "
250 A$(7)=" HELICOPTER IST EIN ABENTEUERSPIEL, BEI "
260 A$(8)="  DEM MAN VERSCHIEDENE AUFGABEN LOESEN  "
270 A$(9)="              MUSS.                      "
280 A$(10)="■                                                   "
290 A$(11)="                                                    "
300 A$(12)="                                                    "
310 A$(13)="                                                    "
```

```
320 A$(14)="    SPEED ┗┛    ┣┫    ┗┛ FUEL    "
330 A$(15)="              ┣┫              "
340 A$(16)="              ┃ ┃              "
350 A$(17)="              ┃ ┃              "
360 A$(18)="              ┃ ┃             ◣ "
370 SP$="                                "
380 HB=1:T1$="000005"
390 ONWBGOSUB1000,2000,2000,3000,3000,4000
400 FORI=1TO3000:NEXT:GOTO390
410 END
1000 REM ---------------
1010 REM > HELI BILD 1 <
1020 REM ---------------
1030 POKE53280,0:POKE53281,0
1040 PRINT"■";
1050 PRINT"■■";
1060 PRINT" ▗▄▄▄
```

```
1190 PRINT"■  ▄■                                                ■▄  ■";
1200 PRINT"■▄■                                                   ■▄■";
1210 FORI=1TO9:PRINTA$(I);:NEXT
1220 FORI=0TO12000:NEXTI
1230 GOSUB10000
1240 ZE=INT(RND(1)*84000+1200):Z4=ZE-1:Z5=ZE+1:TE=ZE-10-INT(RND(1)*50)
1250 Z=ZE:GOSUB11000:ZE$=Z$
1260 Z=Z4:GOSUB11000:Z1$=Z$
1270 Z=Z5:GOSUB11000:Z2$=Z$
1280 Z=TE:GOSUB11000:TI$=Z$
1290 PRINT"DU BIST MIT DEINEM HUBSCHRAUBER AUF DEM ";
1300 PRINT"PARISER FLUGHAFEN. FUER "LEFT$(ZE$,2)":"MID$(ZE$,3,2)":"RIGHT$(ZE$,2)
;
1310 PRINT" ORTS-  ZEIT HAST DU STARTERLAUBNIS. DRUECKE    DANN ";
1320 PRINT"GANZ SCHNELL DIE LEERTASTE!■■"
1330 PRINT"□DIE AKTUELLE ZEIT: "LEFT$(TI$,2)":"MID$(TI$,3,2)":"RIGHT$(TI$,2)
1340 GETA$:IFA$=""THEN1330
1350 ZE$=TI$:HB=HB+1
1360 PRINT"■":GOSUB10040
1370 IFZE$=>Z1$ANDZE$<=Z2$THENPRINT"■■■"TAB(13)"■GUTER START":GOTO1430
1380 IFZE$<Z1$THENPRINT"DU BIST ZU FRUEH GESTARTET!!":GOTO1400
1390 IFZE$>Z2$THENPRINT"DU HAST DEN START VERSCHLAFEN!!"
1400 PRINT"■WENN DU PECH HAST, KOMMST DU EINEM FLUG-ZEUG IN DIE QUERE."
1410 IF.9<RND(1)THENPRINT"■          GLUECK GEHABT!!":GOTO1430
1420 HB=5:TI$="000001"
1430 FORI=0TO4000:NEXT
1440 RETURN
2000 REM ---------------
2010 REM > HELI BILD 2 <
```

```
2020 REM ---------------
2030 POKE53280,0:POKE53281,0
2040 PRINT"■";
2050 PRINT"■ ■■■
```

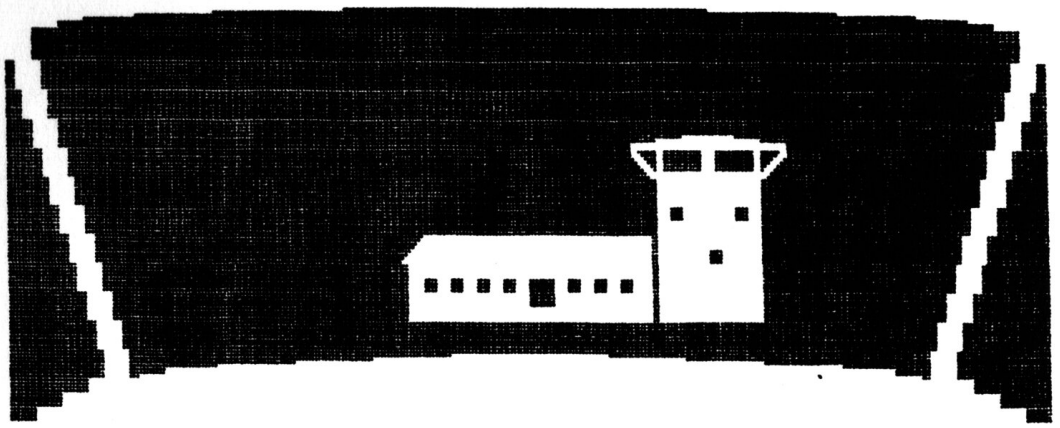
```
2300 DATA"■DIE FREIHEITSSTA- TUE ■ ",NEW YORK
2310 DATA"■ DER ZUCKERHUT ■   ",RIO DE JANEIRO
2320 DATA"■DAS BRANDENBURGER TOR ■ ",BERLIN
2330 DATA"■ DER TOWER ■       ", LONDON
2340 DATA"■ DER PAPSTPALAST■  ",ROM
2350 DATA"■ DIE AKROPOLIS ■   ",ATHEN
2360 DATA"■ DAS OKTOBERFEST ■ ",MUENCHEN
2370 PRINT"FUER WELCHE STADT IST "A1$"DAS WAHRZEICHEN?"
2380 INPUT"STADTNAME";A$
2390 FORI=0TO2000:NEXTI
2400 GOSUB10040
2410 IFA$<>A2$THEN2460
2420 PRINT"BRAVO !"
2430 PRINT"MIT DIESER ORTSKENNTNIS DUERFTEST DU    DICH WOHL KAUM VERFLIEGEN."
2440 HB=HB+1
2450 FORI=0TO5000:NEXT:RETURN
2460 A=0:FORI=1TOLEN(A2$)
2470 IFMID$(A2$,I,1)=MID$(A$,I,1)THENA=A+1
2480 NEXT
2490 IFA/LEN(A2$)<.5THEN2540
2500 PRINT"DU MEINST VERMUTLICH DEN RICHTTIGEN ORT,HAST DIESEN ABER VERKEHRT GE"
;
2510 PRINT"SCHRIEBEN."
2520 PRINT"DIE STADT HEISST RICHTIG ■"A2$"■"
2530 GOTO2440
2540 PRINT"DIE STADT "A$" STIMMT NICHT."
2550 PRINTA1$" LIEGT IN "A2$SP$
2560 PRINT"MIT DIESER ORTSKENNTNIS KANN MAN SICH   LEICHT VERFLIEGEN."
2570 HB=HB+1
```

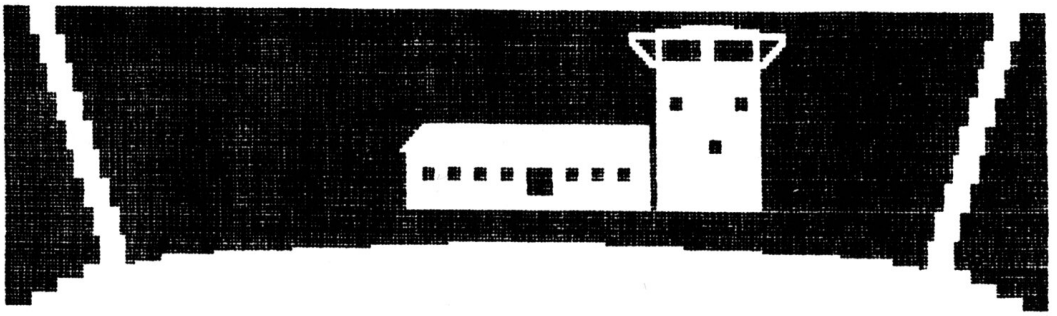
```
2580 IFRND(1)<.91THENHB=5:T1$="000002"
2590 GOTO2450
3000 REM ---------------
3010 REM > HELI BILD 3 <
3020 REM ---------------
3030 POKE53280,0:POKE53281,0
3040 PRINT"■";
3050 PRINT"                                                    ";
3060 PRINT"                                                    ";
3070 PRINT"                                                    ";
3080 PRINT"                                                    ";
3090 PRINT"                                                    ";
3100 PRINT"                                                    ";
3110 PRINT"                                                    ";
3120 PRINT"                                                    ";
3130 PRINT"                                                    ";
3140 PRINT"                                                    ";
3150 PRINT"                                                    ";
3160 PRINT"                                                    ";
3170 PRINT"                                                    ";
3180 PRINT"                                                    ";
3190 PRINT"                                                    ";
3200 GOSUB10020
3210 PRINT"NUN WIRD VON DIR BLITZSCHNELLE REAKTION ";
3220 PRINT"VERLANGT.                               ";
3230 A=INT(RND(1)*6+4):B=INT(RND(1)*8+10)
3240 PRINT"      ┌────"B"──────┐                              ";
3250 PRINT"      |             |                             ";
3260 PRINT"    "A"           "A"                             ";
```

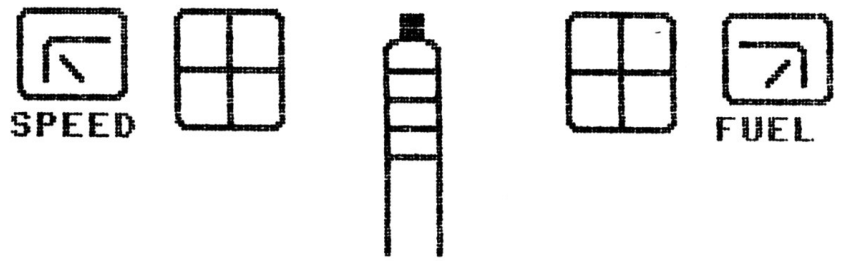
```
3270 PRINT"      |  FLUGZEUG |                                  ";
3280 PRINT" DU  *           <└─> DEINE FLUGBAHN               ";
3290 TI$="000000":T=17:C$="00"
3300 B$="GESAMTFLUGSTRECKE: ":C$="00":GOSUB13010
3310 C=VAL(C$):IFC<>(2*A+B)THENGOSUB13500:GOTO3300
3320 B$="GEFLOGENER UMWEG : ":C$="00":GOSUB13010
3330 C=VAL(C$):IFC<>2*ATHENGOSUB13500:GOTO3320
3340 HB=HB+1:RETURN
4000 REM ---------------
4010 REM > HELI BILD 4 <
4020 REM ---------------
4030 POKE53280,0:POKE53281,0
4040 PRINT"■";
4050 PRINT"■";
4060 PRINT"                                                  ";
4070 PRINT"                                                  ";
4080 PRINT"                                                  ";
4090 PRINT"                                                  ";
4100 PRINT"                                                  ";
4110 PRINT"                                                  ";
4120 PRINT"                                                  ";
4130 PRINT"                                                  ";
4140 PRINT"                                                  ";
4150 PRINT"                                                  ";
4160 PRINT"                                                  ";
4170 PRINT"                                                  ";
4180 PRINT"                                                  ";
4190 PRINT"                                                  ";
4200 PRINT"                                                  ";
```

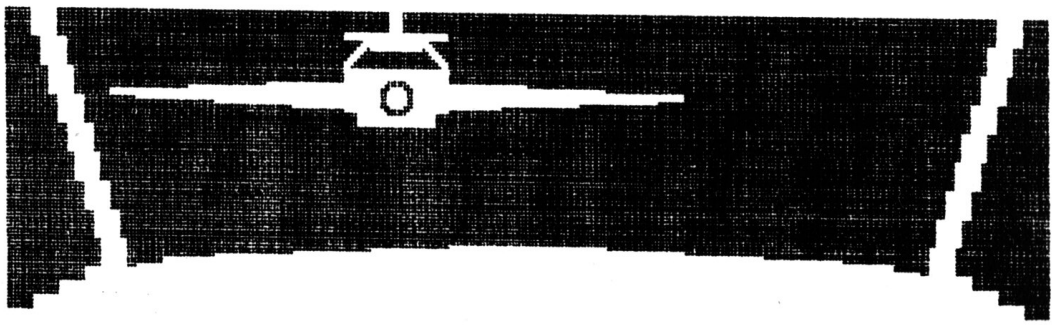
```
4210 GOSUB10020
4220 PRINTTAB(10)"■GUTE LANDUNG!"
4230 PRINT"ICH HOFFE, WIR FLIEGEN BALD WIEDER MIT- EINANDER!"
4240 PRINT"WENN DU NOCH EINMAL FLIEGEN WILLST, DANNGEBE RUN EIN."
4250 END
10000 REM ## AMATUREN AUSDRUCK ##
10010 PRINT"□□□□□□□□□";
10020 FORI=10TO18:PRINTA$(I);:NEXT
10030 FORI=0TO5000:NEXTI
10040 PRINT"□□□□□□□□□";
10050 FORI=1TO9:PRINTSP$;:NEXT
10060 PRINT"□□□□□□□□□";
11000 REM ## ZEITUMRECHNUNG ##
11010 Z1=INT(Z/3600):Z2=INT((Z-Z1*3600)/60):Z3=Z-Z1*3600-Z2*60
11020 Z$=RIGHT$(STR$(100+Z1),2)+RIGHT$(STR$(100+Z2),2)+RIGHT$(STR$(100+Z3),2)
11030 RETURN
12000 REM ## FLIEGENDES FLUGZEUG ##
12010 IFTI$<=T1$THENRETURN
12020 PRINT"□"RIGHT$("■■■
```

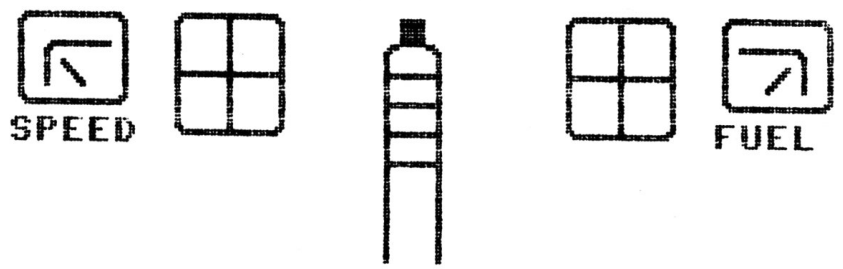
```
13520 POKE53280,16*RND(1):POKE53281,16*RND(1)
13530 FORJ=1TO10
13540 NEXTJ,I
13550 POKE53280,0:POKE53281,0
13560 RETURN
14000 REM ## ABSTURZ ##
14010 I=0:FORJ=2000TO0STEP-2
14020 I=I+RND(1)*J/2:IFI>999.4THENI=I-1000
14030 POKE1024+I,102:POKE55296+I,1
14040 NEXT
14050 FORI=0TO999:POKE1024+I,102:POKE55296+I,1:NEXT
14060 PRINT"■■■■■■■■■■■■■■";
14070 PRINT"■■■■  ■■■  ■■■■  ■■  ■■ ■■ ■■  ■■■  ■■■■";
14080 PRINT"■■■ ■■■ ■■ ■■■ ■■ ■■■■■■ ■■■■ ■■■ ■■ ■■■ ■■■■ ■■■■";
14090 PRINT"■■■    ■■  ■■■■  ■■■■ ■■■■ ■■■ ■■  ■■■■ ■■■■■";
14100 PRINT"■■■ ■■■ ■■ ■■■ ■■■■■ ■■■ ■■■■ ■■■ ■■ ■■ ■■■■ ■■■■■■";
14110 PRINT"■■■ ■■■ ■■  ■■■  ■■■■ ■■■■ ■■■■  ■■■ ■■■ ■■  ■■■■";
14120 PRINT"■■■■■WENN DU NOCH EINMAL VERSUCHEN WILLST,   ";
14130 PRINT"DANN GEBE RUN EIN.                          ";
14140 PRINTSP$SP$SP$"□□□□"
14150 END

READY.
```






SPEED
FUEL




SPEED
FUEL

Notizen über selbstgeschriebene Programme:

Zeilennummer	Anweisung

CIP-Kurztitelaufnahme der Deutschen Bibliothek

Seeßlen, Georg:
Computer kinderleicht / Georg Seeßlen; Peter Fritzsche. –
München; Wien; Zürich: F. Schneider

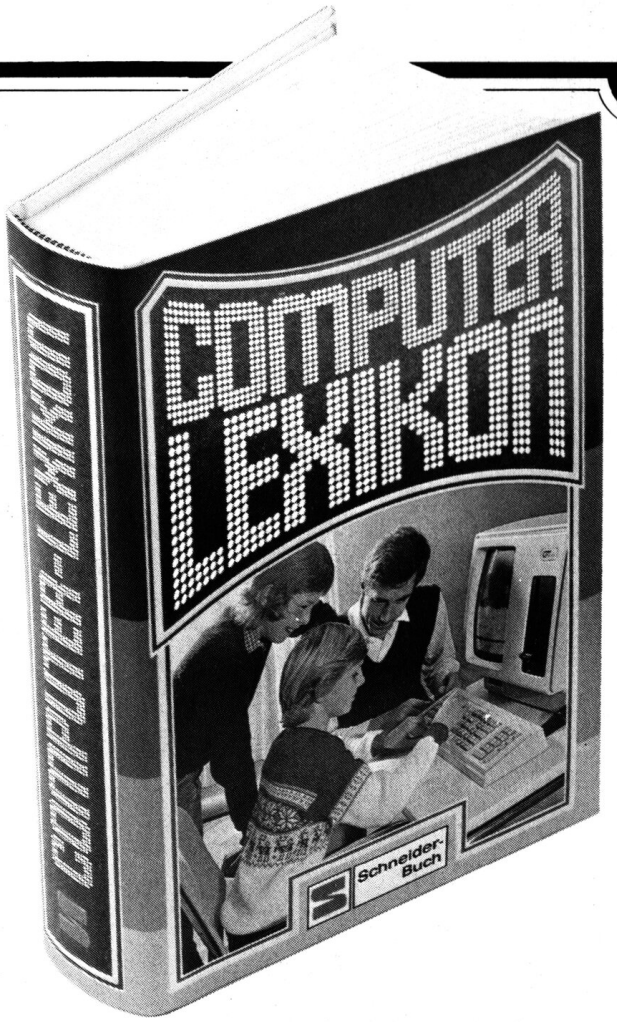
NE: Fritzsche, Peter:

Bd. 1. Wie geht das überhaupt? – 1985.
ISBN 3-505-09090-5
ISBN 3-505-09149-9 Combipack (mit Diskette)

© 1985 by Franz Schneider Verlag GmbH & Co KG
8000 München 46 · Frankfurter Ring 150 · Wien – Zürich
Titelbild und Illustration: Otto Wamser
Lektorat/Redaktion: Thomas de Lates
Redaktion: Thomas de Lates / Otto Wamser
ISBN: 3 505 09090-5
Bestell-Nr.: 9090

Ein weiterer Band ist in Vorbereitung.

Erstmalig:
Mit den Adressen der Hersteller von Kleincomputern



COMPUTER-LEXIKON

Der Computer-Fachmann Klaus Jamin hat dieses Lexikon zusammengestellt. Hier findet der Anfänger und der Fortgeschrittene alles, was er sucht, klar und verständlich erklärt.

- Über 600 Stichwörter von A–Z ermöglichen den direkten Zugriff zum aktuellen Computerwissen unserer Zeit. Sie erklären das Wesentliche und geben präzise Antworten.
- Über 200 Abbildungen informieren ausführlich. Wort und Bild stellen eine Einheit dar, sie ergänzen sich optimal.
- Mit Sonderteil „Programmier-Sprache". Damit nichts verborgen bleibt – für jeden verständlich.

Schneider-Buch

Spannende Schneider-Computer-Bücher

G. P. JORDAN
Computer-Krimi TRM

Band 1 „Geheimprogramm Bienenstich“
Das Geheimnis des verschwundenen Wissenschaftlers

Band 2 „Projekt Gedankenschnüffler“
Was haben die Gangster mit dem Tempest-T vor?

Band 3 „Daten-Piraten“
Tom, Rick und Mona auf Crackerjagd

GEORG SEESSLEN / PETER FRITZSCHE
Computer kinderleicht

Band 1 „Wie geht das überhaupt?“
Einstieg in die Computerei

Band 2 „Lernen Spielen Programmieren“
Der Computer-Rechenkünstler

Wörterbuch für Hacker
Computerslang für Einsteiger

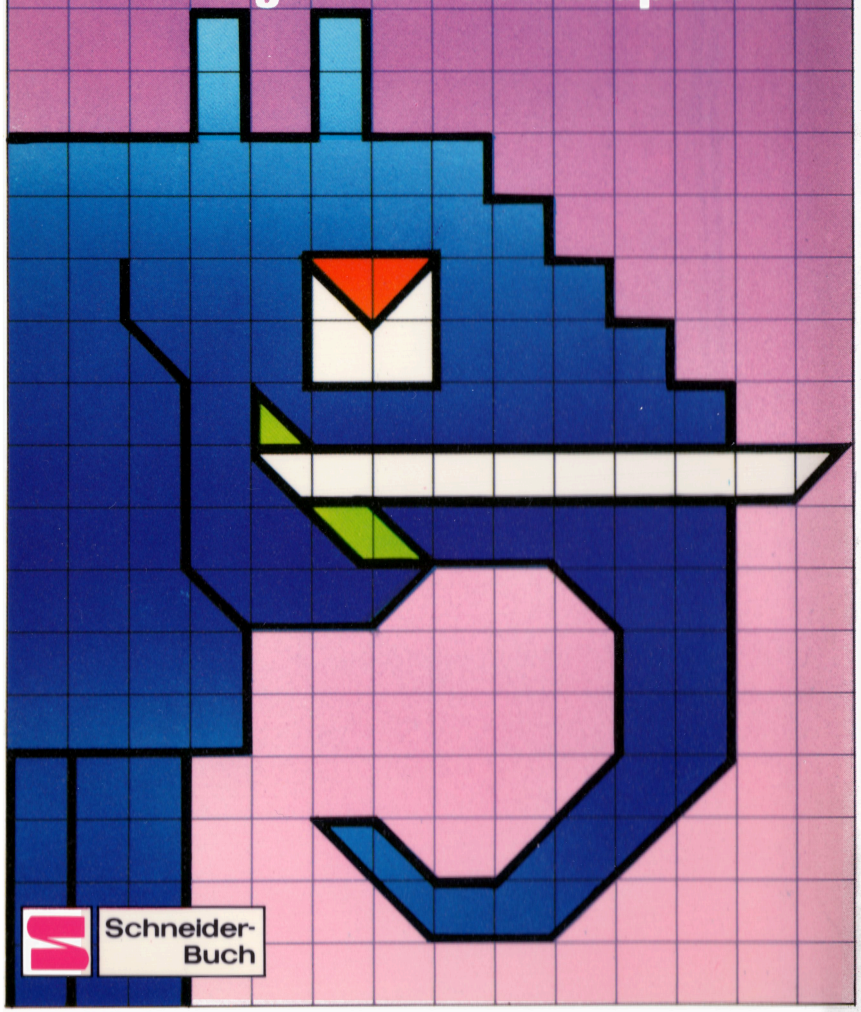
Hacker-Lexikon
Computerslang für Insider

KLAUS JAMIN
Computer-Lexikon
mit Hersteller-Adressen
und Sonderteil „Programmiersprache“

KLAUS JAMIN
Computer-Tagebuch
mit Programmiertips und einem witzigen Programm für den C 64

Computer kinderleicht

Wie geht das überhaupt?



Schneider-Buch

COMPUTER

SEESSLEN/FRITZSCHE · Computer kinderleicht

Bd. 1

Computer kinderleicht · Bd. 1
Wie geht das überhaupt?

„Computer kinderleicht" – das heißt computern lernen ohne Mißerfolge, ohne „schwarze Löcher" und ohne Langeweile. Geschrieben von Computer-erfahrenen Pädagogen unter kritischer Beteiligung von Computer-neugierigen und schließlich Computer-begeisterten Kindern.

Band 1 enthält das Wichtigste für den systematischen und schöpferischen Einstieg in die Computerei, verständlich und unterhaltsam, mit Erfolgserlebnissen von Anfang an! Ein Buch mit zahlreichen Abbildungen und Programmlistings, das Kinder eigenständig und weitgehend ohne Hilfe von Erwachsenen benützen können.

9 783505 090905

ISB N 3-505-09090-5